BOURNEMOUTH UNIVERSITY

TRANSFER DOCUMENT

From Brush Painting to Bas-relief

Author:
Yunfei Fu

First Supervisor:
Dr. Hongchuan Yu
Second Supervisor:
Pro. Jian Jun Zhang

July 3, 2017

ABSTRACT

In this research , we present a new approach for generating bas-reliefs from brush paintings. Our approach exploits the concept of brush strokes, making strokes possible to generate 3D proxies separately suitable for recomposing in art design. To segment brush strokes in 2D paintings, we reformulate layer decomposition and MSERs segmentation by imposing boundary constraints, which can make strokes smooth and complete. The resulting 3D proxies of brush strokes are sufficient to evoke the impression of the consistent 3D shapes, so that they may be further edited in 3D space.

As showed in Figure 1, the approach performs in three steps: First, based on the point cloud of the input image in RGB space, we select the palette colors of the input brush painting, and based on the palette colors we decompose a input image into different layers with transparency, Second, the original painting is decomposed into element brush strokes by applying modified MSERs segmentation.Based on our observation, alpha map is more suitable for our MSERs segmentation, so the segmentation is based on the transparency. Third, based on the segmented brush strokes, Shape from Shading algorithm has been applied to generate the depth map of each stroke. finally, based on those 3D strokes , we can edit the generated bas relief. For editing,we have already generated the 3D proxies for each MSER region, namely, the extracted 2D brush region, so, we can change the depth of specific stroke on bas relief,we can also stitch them on each other based on the user input and change the shape of a certain stroke with given indicated skeleton. By doing so, we fulfill the request of recomposition in bas-relief design.

We make several technical contributions: 1. Extend the novel algorithm for MSER. 2. Stroke extraction from different layer based on a single image. 3. Detail preserving 3d shape reconstruction from brush sketch.

We demonstrate our approach on a variety of brush paintings, showing that even in different styles of brush paintings, e.g. Chinese brush paintings and North European brush paintings, our approach is able to produce convincing bas-reliefs. We demonstrate that the proposed approach is particularly suitable for converting real brush painting into plausible bas relief shape with minimum interaction.

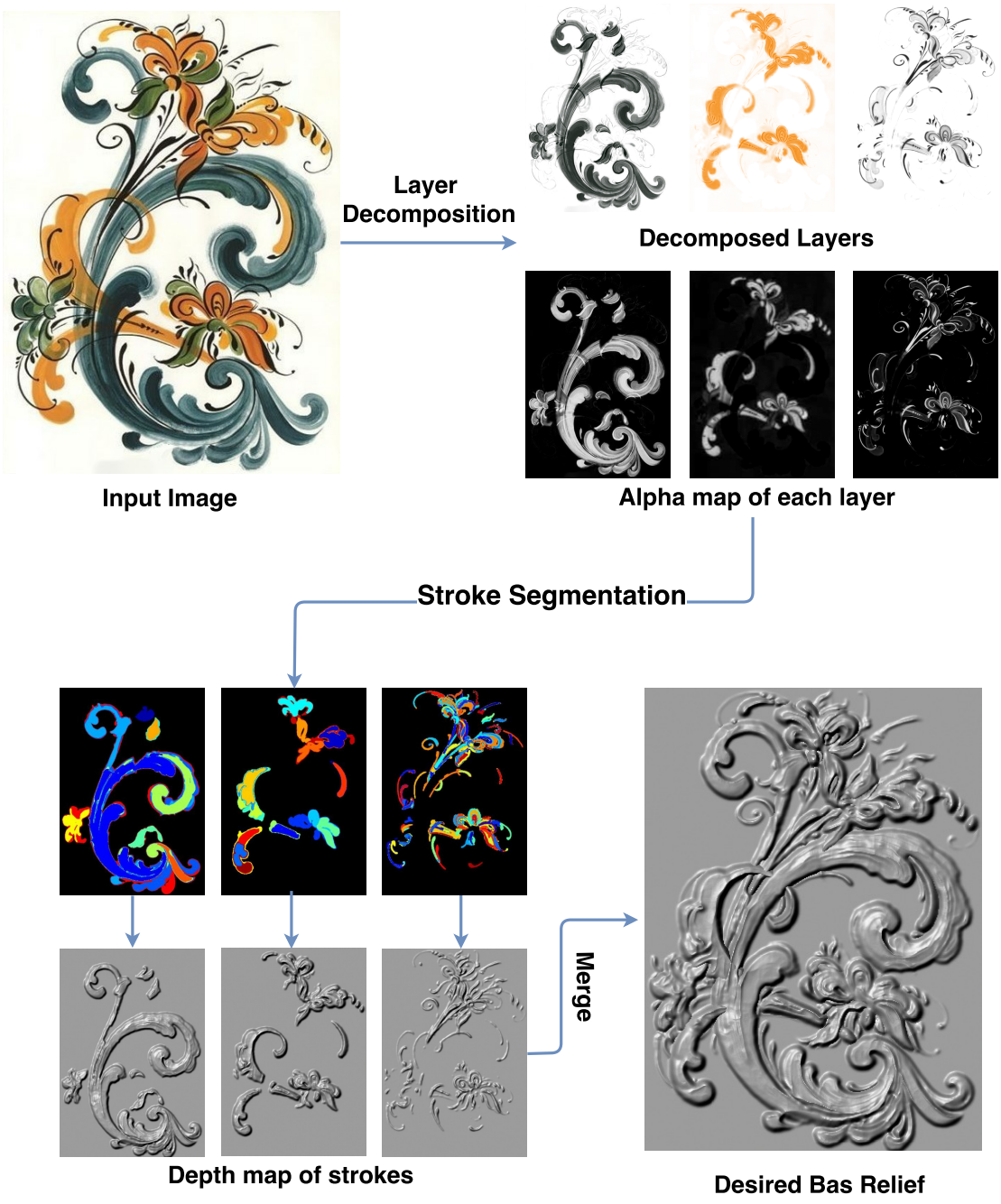


Figure 1: Pipeline Overview

Contents

List of Figures

Chapter 1

INTRODUCTION

Relief is a method of sculpture in which forms are carved into a relatively flat surface. Essentially it creates a bridge between full 3D sculpture and 2D media e.g. from images to sculpture. Relief sculpting has been practiced for thousands of years.Since antiquity, artisans from many ancient cultures(including Greek, Persian, Egyptian, Mayan, and Indian art) have created bas-reliefs (Figure 1.1) .Today bas-reliefs are commonly found in a variety of media in architecture, industrial design and coins. However, The production of bas-reliefs is currently a costly and timeconsuming process, requiring skilled sculptors and engravers.Even with the advent of computer-driven techniques providing a foundation for automation in bas-relief, making the design of bas-relief sculptures remains largely in the hands of artists.

As an artistic form, relief spans the continuum between a 2D drawing or painting and a full 3D sculpture[1]. On this spectrum, alto-relievo (high relief) is closest to full 3D, whereas flatter artworks are described as basso-relievo (low relief, and also called bas-relief), as showed in Figure 1.1. Among all the sculpture forms, bas-relief is arguably the closest to 2D paintings, as claimed by[2] and[3].

Figure 1.1: Bas-relief

Currently, most existing bas-relief production methods focus on compressing 3D scenes/models into surfaces with a small depth[1]and[2]. This approach requires a 3D model as a starting point. Another option is to generate bas-reliefs directly from 2D images. Examples can be found on the coins with the profiles of some US presidents and the queen of the UK, which are produced by sculptors. There have been some image based bas-relief production approaches available in[4][5] and [6]. Unlike the 3D model based methods, spatial

occlusion (i.e. one stroke occludes other strokes in the paining to demonstrate the depth perception) is not taken into account. These approaches almost follow the "bas-relief ambiguity"[7], that is, roughly speaking, from a frontal view the sculpture looks like a full 3D object while a side-view reveals the disproportional depth. It offers the smallest distortion from the frontal view. To reduce visual distortion, one of the criteria for the image-based approach is perspective geometry preservation. Bump mapping [8] [9] provides the ability to create a bas-relief effect, which has been involved in some modern image and graphic processing software, such as ZBrush and Photoshop. The basic idea is to make a rendered surface look more realistic by modifying the surface normal. If the height map is available, the normal is calculated and modified directly. Otherwise, the normal map may be generated through 2D grayscale or the gradient of the texture images. Bump mapping does not modify the surface geometry. Although the image-based methods eliminate the need of a 3D model, a raising issue is how to generate bas-reliefs from brush paintings. A clear shortcoming is such methods can't take the artistic intent into account, as all what they do is to inferring the height information from the image. Concerning reproduction or repurpose of an artistic painting, it is crucial that the style of the originals is preserved, which is not considered in existing image-based methods. However, little is done in the area of bas-relief generation from artistic paintings, as maintaining the styles of the brush paintings proves much trickier than simply manipulating the height of the contour lines. In the case of bas-reliefs, although there is no 3D model available, pseudo 3D effect reflecting the style and subtlety is crucial in preserving the artistic essence. The aim of our research is to provide the bas-relief sculptors with a new tool allowing them to convert and recompose existing brush paintings to bas-reliefs. We also argue that because traditional paintings are produced with individual strokes, '3D strokes' will enable them to 'paint/sculpt' a bas-relief naturally, especially if they wanted to quickly convert an existing painting into a relief. With the commonly and cheaply available 3D printing facilities, there is a growing trend in the need of bas-relief art products. A brush painting can be regarded as the union of a set of hypothetical strokes by a brush [10]. Differing from the other bas-relief generation methods, our method will honor this very feature by constructing the brush strokes individually as 3D geometric entities. This however demands to conquer several challenges. First, each brush stroke covers a region on the canvas and they may overlap each other, some quite heavily in a painting. To make sure the information is retained, every stroke has to be faithfully extracted. Second, spatial occlusion has to be dealt with, since artists are used to depicting it through controlling the transparency of strokes as one of the art elements. Third, as an artistic tool, the generated bas-relief should be further editable allowing the artist to rearrange, tweaking and reshaping the extracted strokes. The shape, color and opacity of a stroke vary due to the shape and firmness of the brush as well as the forces the artist imposes. Although these variables add the complexity to stroke extraction, stylized strokes often follow distinct patterns. For example, Rosemaling paintings, a typical example of brush painting popular in North Europe, is a traditional form of decorative folk art that originated in the rural valleys of Norway. The Rosemailing designs use C and S strokes, feature scroll, flowing lines, floral designs, and both subtle and vibrant colors. The brush strokes may further be viewed as graphical objects which are meaningful with respect to the objects the painting portraits. Moreover, each stroke is clearly visible due to both subtle and vibrant colors. The similar properties may be found in some Chinese brush

paints and oil paintings such as those of van Gogh. To extract the strokes from a brush painting, we need to identify and segment the overlapped strokes. We will then generate the depth map for every stroke separately using the shape from shading (SFS) technique on the opacity. All the strokes are finally merged together to yield the resulting bas-relief with the original 2D painting preserved. Our contributions include,
(1) Extraction of brush strokes. We develop a novel method to extract brush strokes from input paintings with palette analysis and decomposed layers.
(2) 3D modeling of brush strokes. We develop a novel method which may entirely construct every stroke in 3D based on the opacity of paintings.
(3) Recomposition in art design. Artists may redefine the brush strokes' order and shapes by sketches, which enable recomposition in bas-relief design, making it a useful tool for sculpture artists.

1.1 Thesis Structure

The organization of the document is as follows:

Chapter 1: Introduction. This section provides the background, the motivation and the contribution made in current research.

Chapter 2: Literature Review. This section classifies and reviews related previous works in a systematic way.

Chapter 3: Overview of proposed approach. This section explains the methodology selected and defines some basic concepts used in the research.

Chapter 4: Layer Decomposition. This section introduces concept "Layer Decomposition" in this research, which is the basis of the proposed algorithm.

Chapter 5: Extraction of Brush strokes. This section describes the proposed algorithm of brush stroke extraction.

Chapter 6: Depth map and '3D strokes'. This section shows how to generate the individual depth maps of extracted strokes.

Chapter 7: Results. This section reviews the up-to-date results based on proposed the method.

Chapter 8: Conclusions . This section concludes the progress up-to-date.

Chapter 9: Future work. This section discusses possible future work about high relief and 3D painting.

Chapter 2

LITERATURE REVIEW

Inferring 3D shape from images has a wide range of applications for computer vision, graphics, object recognition, surveillance, and HCI. This is a highly under-constrained problem, which requires as much as possible prior knowledge of the 3D shape of an object class so as to yield reasonable results [11]; [12]. When the input is a man-made painting or sketch, it becomes more challenging. This is because (1) the man-made paintings may violate the perspective geometry; (2) the line drawings may be ambiguous; (3) the input may consist of a lot of strokes that need to be specified by the users; and (4) these strokes are usually inter-related for artistic composition purposes. The closest approach to the topic of brush painting based bas-relief production is the image based method. [6] proposed a discrete grid for approximating the illumination of a given image. The key point is to approximate the Lambertian Reflection model by a linear system, which may result in a distortion of the frontal view. In contrast, [5]preferred to standard lighting conditions so as to reduce distortion. Hence, the input face images have to be relit under standard lighting conditions through offline learning techniques.[4]applied the method of Poisson surface reconstruction to bas-reliefs. To make a layered look, the depth map has to be post-processed. Since inputs are usually images of real scenes or faces, perspective distortion becomes a critical issue. Moreover, these methods do not take into account occlusion, which painters are just most concerned with. frequently used by painters in paintings. Therefore, they are not suitable for painting based bas-relief production. There have been other works on 3D reconstruction from 2D line drawings[13];[14];[15]; [16] . However, our research aims at paintings with brushes, which are different from 2D line drawings. A stroke does not only contain lines but also delineates a region. It is crucial to identify and extract brush strokes from a painting.

2.1 Stroke Segmentation

To the best of our knowledge, there is lack of study on extracting brush strokes from paintings. We give a brief overview of the work related to the relevant topics, i.e. decomposing images into layers and stroke segmentation, which are employed in our implementation. In digital image editing, layers organize images. However, scanned paintings and photographs have no such layers. Without layers, simple edits may become very challenging.[17] presents an approach to produce editable vector graphics, in which the

selected region is decomposed into a linear or radial gradient and the residual, background pixels. [10] aims at decomposing Chinese paintings into a collection of layered brush strokes with an assumption that at most two strokes are overlapping and there is minimally varying transparency. [18],[19]present two generalized layer decomposition methods, which allow pixels to have independent layer orders and layers to partially overlap each other.[20] present a layer decomposition method based on RGB-space geometry. They assume that all possible image colors are convex combinations of the paint colors. Computing the convex hull of image colors and per-pixel layer opacities is converted into a convex optimization problem. Thus, their method can work well without prior knowledge of shape and overlap of strokes. Currently, most research focuses on stroke segmentation of handwritten characters, such as pen strokes[21]. Pen stroke edges are distinct and can be extracted entirely in handwritten characters. As a result, the basic idea is to describe a stroke by a set of geometric primitives, so that the hand-drawn primitives may be replaced by mathematically precise shapes to produce a neat final result. However, brush strokes on a painting often contain individual colors and overlap each other. In fact, the strokes may show a low contrast to the others or the background. The edges of strokes are blurred.[10] extracts descriptions of the brush strokes by using a brush stroke library. However, their approach requires a good amount of prior knowledge of shape and order of strokes. The Most Stable Extremal Regions (MSERs) algorithm[22] was used for establishing correspondence in wide-baseline stereo.[23] introduced the data structure of the component tree in it and further developed it as an efficient segmentation approach, which prunes the component tree and selects only the regions with a stable shape within a range of level sets. The revised version has been widely used in stroke segmentation of handwritten characters [24]

2.2 Shape from shading

Shape recovery is a classic problem in computer vision.The goal is to derive a 3D scene description from one or more 2D images. The computation process normally involved with several concepts : depth $Z(x, y)$, surface normal n_x, n_y, n_z ,and surface gradient p, q. The depth can either be considered as distance from viewpoint to query surface or the height from surface to default $x - y$ plane. The normal is perpendicular to the surface gradient, namely $(n_x, n_y, n_z) * (p, q, 1)^T = 0$, the surface gradient is the changing rate of depth in x and y direction.

Shape from shading (SFS) is able to recover the shape of an object from a given single image, assuming illumination and reflectance are known (or assuming reflectance is uniform across the entire image). Many methods have been developed, which may be categorized into four PDEs models[25], (1) orthographic SFS with a far light source [26]; (2) perspective SFS with a far light source[27]; (3) perspective SFS with a point light source at the optical center[25]; (4) a generic Hamiltonian. However, SFS is an ill-posed problem. The notable difficulty in SFS is the bas-relief ambiguity[7], that is, the absolute orientation and scaling of a surface are ambiguous given only shading information. To amend it, many SFS algorithms impose priors on shape, depth cues produced by a stereo system, or assume that the light source, the surface reflectance, and the camera are known[28]; [29]; [30]; [3]; [31]. However,[32] argued that "the knowledge of accurate depth values is not

necessary to produce believable advanced illumination effects". They further presented an approach for generating global illumination renderings of hand-drawn characters through a bas-reflief type mesh. The key point is to create such a mesh from a hand-drawn image using only the contours and the layering relationships of the components. Unfortunately, the resulting mesh looks inflated since it is generated only by Dirichlet and Neumann boundary constraints.

To recover the shape of scene from shading of single image,it's important to exploit how image formed from lighting and shading.One simplest model is Lambertian shading model, in which the intensity of image are purely depends on the normal of surface and light direction. Given the intensity of each pixel of image, SFS aims to recover the light direction and surface shape. Meanwhile, in real world, situation is more complex than the Larbertian shading model,even with known light direction, and the gray level can be described as a function of surface shape and light source direction, the problem is still not simple.

If the surface shape is described in terms of the surface normal, we have a linear equation with three unknowns, and if the surface shape is described in terms of the surface gradient, we have a nonlinear equation with two unknowns. Therefore, finding a unique solution to SFS is difficult; it requires additional constraints.

To solve the SFS problem, there are four common ways: minimization approaches, propagation approaches, local approaches, and linear approaches [28]. Minimization approaches, namely, focus on how to form a energy function and obtain the solution by minimization. Propagation starts certain points of the surface. Local approaches based on assumption of surface type. Linear approaches compute a linearization from reflection model.

2.2.1 Minimization Approaches

Ikeuchi and Horn [33] proposed the method to recover the surface gradient. Since each surface point has two unknowns for the surface gradient and each pixel in the image provides one gray value, we have an underdetermined system. To solve this problem, the brightness constraint and the smoothness constraint were introduced . The brightness constraint aims to produce the shape that generate the same brightness of input image, and the smoothness constraint constrains the smoothness of reconstructed shape. By minimizing these two constraints,the shape was computed. To ensure a correct convergence, the shape at the occluding boundary was given for the initialization.Because on the boundary of objects the gradient is infinite, stereographic projection was also applied to transform the the error function to a different space.Using these two constraints, Brooks and Horn [34] also use minimization method,in terms of the normal of surface. The integrability was enforced in [35] to generate integrable surface.

Surface slope estimates from the iterative scheme were expressed in terms of a linear combination of a finite set of orthogonal Fourier basis functions. The enforcement of integrability was done by projecting the nonintegrable surface slope estimates onto the nearest (in terms of distance) integrable surface slopes. This projection was fulfilled by finding the closest set of coefficients which satisfy integrability in the linear combination. Compared with the earlier research, their method is more accurate and efficient. Szeliski [36] applied a hierarchical basis preconditioned conjugate gradient descent algorithm .To

improve the stability of Brooks and Horn's algorithm a heuristics also was applied to the variational approach by [37].

Intensity gradient constraint[38] was introduced to constraint the difference of gradient between the reflect map and input image in both x and y direction , which was used to substitute the smooth constraint.

While the above mentioned method are based on variational calculus. [39] focusing on calculate depth based on discrete formulation and conjugate gradient method. To achieve convergence, brightness constraint and smoothness constraint should be applied. Lee and Kuo [39] proposed a method does not need to initialization of depth,and they approximated surfaces by a union of triangular patches.

Above mentioned approaches are based on producing a single smooth surface. Malik and Maydan [40] focused on piecewise smooth surfaces. To minimize the energy function, line drawing and shading constraints were applied, and by such, they finally recover both surface normal and line labeling.

2.2.2 Propagation Approaches

Horn proposed the characteristic strip method [41] which is based on propagation.In characteristic strip method, if the at the beginning point of the characteristic strip line , surface depth and orientation is already known, then along the line all the other points along the line can be computed. Characteristic strip method starts with singular points, pixels with maximum intensity, and construct their neighborhoods based on sphere approximation which results in some initial surface curves. The direction of the strips are determined by the local gradient , and along the characteristic strips the depth information would be propagated outwards. Since only by the singular points, the correspond curves are relatively sparse, to produce dense shape, new characteristic strips were interpolated from the initial strips.

Hamilton-Jacobi-Bellman equations and viscosity solutions theories was applied in the work of Rouy and Tourin [42] , and by such assumption, we can have an unique solution. Some other researches proves that surface reconstruction can starts from singularity points instead of occluding boundaries [43], and based on such idea, Shape from shading can be formulated as the optical control problem , and it can be solved by numerical methods [44]. Based on Dupuis and Oliensis's approach, a minimum downhill method shows a improvement on efficiency which could achieve convergence in 10 iteration [45] .

Similar with these methods, starting with a close curve in the areas of singular points, Kimmel and Bruckstein [46] proposed a method based on layers of equal height contours to reconstruct the surface.By applying differential geometry, fluid dynamics, and numerical analysis, their outcome enabled nonsmooth surface generation.

2.2.3 Local Approaches

In the work of Pentland [47], the first and second derivatives were applied for the surface reconstruction and base on the assumption of every local points is a spherical. Similarly, under this locally spherical assumption, and by using a light surce coordinate system, slant and tilt of the surface can be computed [48] from the first derivative of the image.

2.2.4 Linear Approaches

Some works mentioned above are based on linear approaches which aim to reconstruct the depth map from a linearization based on the reflectance. By Fourier transfer the linear function can have a closed from solution at each point of depth [49] , and we can have such a linear approximation of the reflectance map from local gradient. Jacobi iterative scheme was also been used for recovering the surface at each pixel in the work of Ping-sing and Shah [50],and they used the discrete approximation of the gradient first, then employed the linear approximation of the reflectance function in terms of the depth directly.

Interreflections

While above mentioned researches simply recover the surface based on the reflect function,in the works of Nayaret et al. [51] [52] inter-reflection has also been put into consideration. Based on their observation, the shape reconstructed from shape-from-photometric-stereo algorithms has erroneous, in the area which effected by inter-reflection, the recovered shape is shallower than the real shape. So,their method focused on this problem based on iteratively refinement. Similar method was proposed by Forsyth and Zisserman [53].

Convergence

Some researches focus on the convergence of the SFS function, based on the work of Oliensis [43] , the solution has uniqueness when the light source direction is equal to or symmetric to view source. If the first derivative of the surface is continuous, the characteristic strip could yield a unique solution. Commonly, the uniqueness of SFS problem is unknown. But in the work of Lee and Kuo [39], with given depth of some points, and constrains of the smoothness, they can obtain a convergence from most cases based on their linearization of the reflection function.

2.3 Reflectance Models

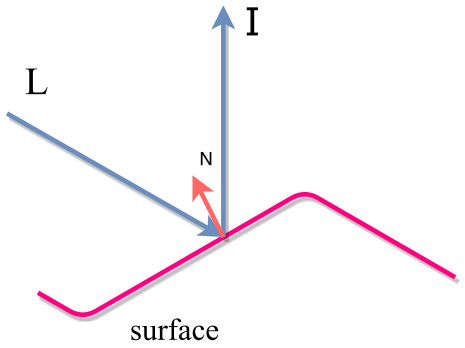


Figure 2.1: Reflectance

Reflectance is a physical property of a material that describes how it reflects incident light. The appearance of various materials are determined to a large extent by their reflectance

properties. Most reflectance models can be broadly classified into two categories: diffuse and specular. In computer vision and computer graphics, the diffuse component is often assumed to be Lambertian.

Lambertian reflectance:

Lambertian reflectance is the property that defines an ideal "matte" or diffusely reflecting surface. The apparent brightness of a Lambertian surface to an observer is the same regardless of the observer's angle of view. More technically, the surface's luminance is isotropic, and the luminous intensity obeys Lambert's cosine law. So the reflectance map can be model as a production of Light and normal of the surface, the albedo of surface ρ . where α is the angle between the directions of the two vectors, the intensity will be the highest if the normal vector points in the same direction as the light vector

$$I_L = L * N = |L||N|cos(\alpha) \tag{2.1}$$

where α is the angle between light direction L and normal of the surface N. Although the Lambertian model is widely used because of its simplicity, it is a poor approximation to the diffuse component of rough surfaces.

Specular reflection

Specular reflection, also known as regular reflection is the mirror-like reflection of waves, such as light, from a surface. In this process, each incident ray is reflected, with the reflected ray having the same angle to the surface normal as the incident ray. The result is that an image reflected by the surface is reproduced in mirror-like (specular) fashion. The law of reflection states that for each incident ray the angle of incidence equals the angle of reflection, and the incident, normal, and reflected directions are coplanar. Specular reflection is not considered in our research.

Chapter 3

OVERVIEW OF PROPOSED METHOD

As shown in Figure 3.1, a given painting is firstly decomposed into a set of layers in terms of RGB values. Secondly, each resulting layer is further segmented into multiple regions which represent brush strokes respectively. Thirdly, the depth map of each brush stroke is generated by the shape from shading techniques individually. After that, the desired bas-relief is generated by merging all the depth maps of brush strokes together.
The key point is to extract brush strokes from input paintings. Overlapped strokes make colors blend. To deal with it, layer decomposition is firstly employed here, which decomposes the painting into a set of single colored and translucent layers. In brush paintings, each stroke only utilizes a single palette color. Therefore, layer decomposition helps classify brush strokes separately into different layers based on the palette color so that every layer contains the strokes which are well separated. However, wrong layer decomposition may cut one stroke into two or more layers. It is observed that typically strokes of such paintings follow regular patterns. For instance, Rosemaling paintings employ many C and S strokes, and the color and transparency change very little in the direction of the stroke. We introduce the edge tangent flow (ETF) field and the coherent line [54] to enhance such features in paintings, which is in favor of preserving the wholeness of the strokes in every layer and effectively avoid wrong layer decomposition.
The coherent line is further involved in the MSERs algorithm [23] again for extracting spurious edges within one layer. Furthermore, to generate the depth maps of the strokes

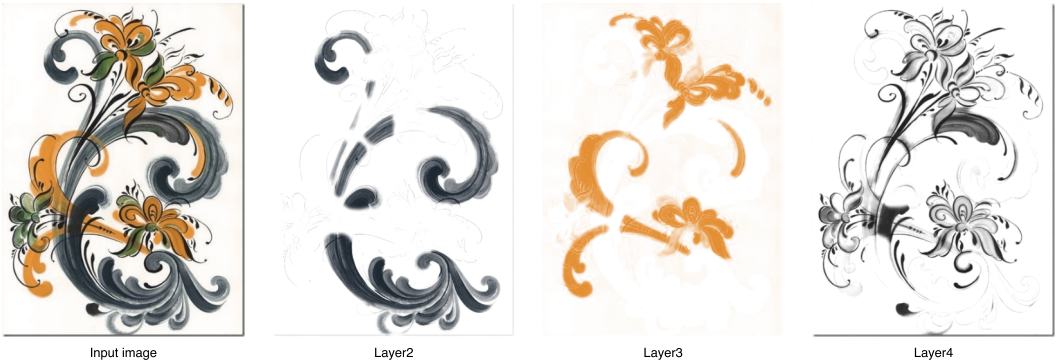


Figure 3.1: Decomposition layers

individually, we perform shape from shading on the opacity of the paintings instead of the intensity here, since the opacity has a bigger range than the intensity (as can be seen in Figure 4.9 for detail). SFS techniques may generate details of surfaces in terms of image texture. It is desirable to transfer the features of the paintings, e.g. the density of colors, to the surface of the brush stroke models.
The depth maps of all the strokes are then merged together to form the desired bas-relief. We hope to point out that the employed orthographic SFS technique tends to encourage interior growth within a closed region due to the Eikonal Equation, which may result in the growth of a background region surrounded by strokes. In general, the background plane should be unchanged. Thus, generating strokes individually not only benefits the composition of bas-reliefs but also avoid this technique issue. Moreover, the stroke order and shape may be edited by users to cater for the request of recomposition in art design. The user may recompose the stroke models in 3D space for secondary creation.

Chapter 4

LAYER DECOMPOSITION

Digital painting with different layers is an integral feature of digital image editing software, such as Photoshop and Sketchbook. Layers offer an intuitive way to edit the color and geometry of components and localize changes to the desired portion of the final image. Without layers, brush stroke segmentation becomes extremely challenging, since they can overlap and blend with each other.

In general, each layer represents one coat of a painting with single color that is applied with varying opacity throughout the input painting. Wrong layer decomposition may cut one stroke into different layers. It is crucial to preserve the completeness and smoothness of the brush strokes in layer decomposition. To this end, we modify the layer decomposition algorithm in by involving the coherent lines [54] in our implementation. In the following we first address the layer decomposition algorithm briefly and then discuss our modification.

4.1 Identify Paint Colors

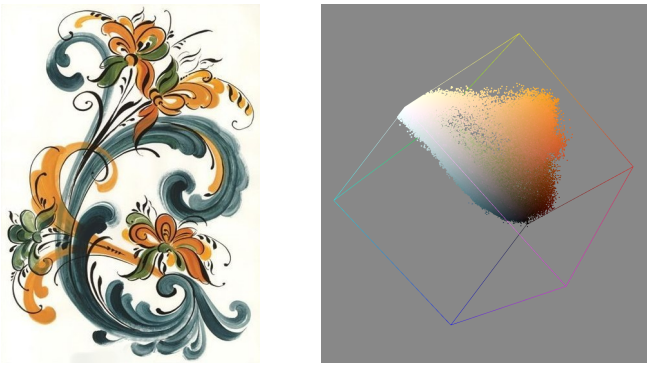


Figure 4.1: Geometry of input image pixels in RGB-space

In a brush painting, one region may have been painted multiply time with different paint colors. We assume that the color on each pixel is a linear combination of the paint colors, so all the pixel color in the input painting lies the convex hall of in the RGB space as showed in Figure 4.1. And base on such idea, we can represent each pixel color based on painted color:

$$p = \Sigma \omega_i c_i$$

p represents the color of the pixel, and c_i represents the i-th paint color. To compute the paint color, we introduced the convex hull simplifying method of Tan's work [20]. In which a convex hull of the colors in RGB space should be computed, while every vertice is considered as a paint color. The colors would be tightly wrapped by the convex hull, but normally there would be many vertices more than what we need, since too many vertices would result in too many layres. In Tan's work they provide a simplification method which would output manageable number of layers based on user need and the output layers with clearly differentiated colors,as showed in Figure 4.2.

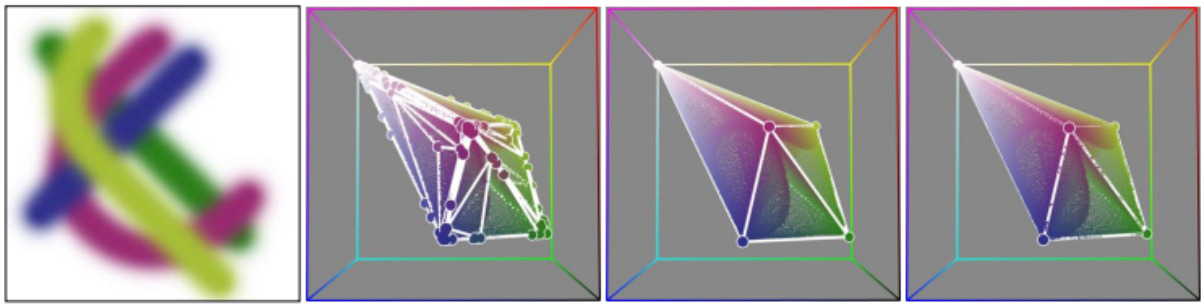


Figure 4.2: Convex hull of input image

(a) A simple digital painting's (b) convex hull in RGB-space is complex due to rounding. (c) The result of simplification algorithm (d) matches ground truth, its original paint colors as an RGB-space polyhedron.

4.2 Layer Decomposition Scheme

The "A over B" compositing and blend mode in [55] is described that when the pixel A with color A_{RGB} and translucency α_A is placed over the pixel B with color B_{RGB} and translucency α_B, the observed color is,

$$\left(\frac{A}{B}\right)_{RGB} = \frac{\alpha_A A_{RGB} - (1-\alpha_A)\alpha_B B_{RGB}}{\left(\frac{A}{B}\right)_\alpha}$$

where

$$\left(\frac{A}{B}\right)_\alpha = \alpha_A + (1-\alpha_A)\alpha_B$$

Each pixel's color is viewed as the convex combination of all layers' colors. For each pixel, the observed color p can be approximated by the recursive application of the compositing and blend model. We take as input ordered RGB layer colors through computing per-pixel opacity values for each layer. The following 'polynomial' regularization term penalizes the difference between the observed color p and the polynomial approximation,

$$E_{polynomial} = \frac{1}{K}\left\| C_n + \sum_{i=1}^{n}\left((C_{i-1} - C_i)\prod_{j=i}^{n}(1-\alpha_j)\right) - p\right\|^2$$

where C_i denotes the i-th layer's color, is the opacity of α_i , the background color C_i is opaque, $K = 3$ and or 4 depending on the number of channels (RGB or $RGB-\alpha$).

The opacity penalty is expressed as,

$$E_{opaque} = \frac{1}{n}\sum_{i=1}^{n} -(1-\alpha_i)^2$$

The default smoothness penalty is expressed as,

$$E_{opaque} = \frac{1}{n}\sum_{i=1}^{n} (\bigtriangledown\alpha_i)^2$$

where $\bigtriangledown\alpha_i$is the spatial gradient of opacity in the i-th layer. This term penalizes solutions which are not spatially smooth. However, the gradient of opacity is not always aligned with that of intensity, which may result in edges discontinuous. The users may specify the layer order in advance, as well as the number of layers, n, is given. The opacity for every layer may be solved by minimizing the following combined cost function,

$$E = \omega_{polynomial}E_{polynomial} + \omega_{opaque}E_{opaque} + \omega_{spatial}E_{spatial} \tag{4.1}$$

where $\omega_{polynomial} = 375, \omega_{opaque} = 1, \omega_{spatial} = 10$.

4.2.1 Modified Layer Decomposition

Figure 4.3: Edge Tangent Flow

To enhance the smoothness and completeness of edges, the coherent line drawing technique in [54] is introduced to Eq 4.1 ,which is a flow-guided anisotropic filtering framework. Figure 4.3 shows the edge tangent flow (ETF) field of a Rosemaling painting. First, the ETF field is involved in $E_{spatial}$.
The ETF field is defined as,

$$t^{new(x)} = \frac{1}{k}\sum_{y\in\Omega(x)} \varphi(x,y)t^{current}(y)\omega_s(x,y)\omega_m(x,y)\omega_d(x,y) \tag{4.2}$$

As showed in Figure 4.4, $t(x)$ denotes the normalized tangent vector at pixel x, $\Omega(x)$ denotes the neighborhood of the pixel x, and k is the term of vector normalization. The

spatial weight function ω_s employs the radially-symmetric box filter with some radius. The magnitude weight function ω_m is monotonically increasing, indicating that the bigger weights are given to the neighboring pixels y whose gradient magnitudes are higher than that of the central pixel x. This ensures the preservation of the dominant edge directions. The direction weight function, ω_d, may enhance alignment of vectors, e.g. $t(x) \cdot t(y) > 0$, while suppressing swirling flows. In addition, the sign function $\varphi(x, y)$ is employed to prevent the swirling artifact as well.

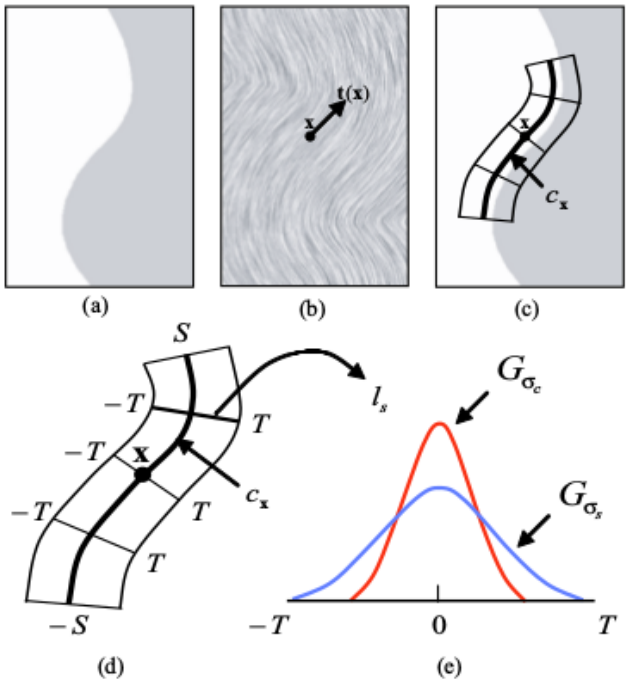


Figure 4.4: Coherent line details

(a) Input (b) ETF (c) Kernel at x (d) Kernel enlarged (e) Gaussian components for DoG

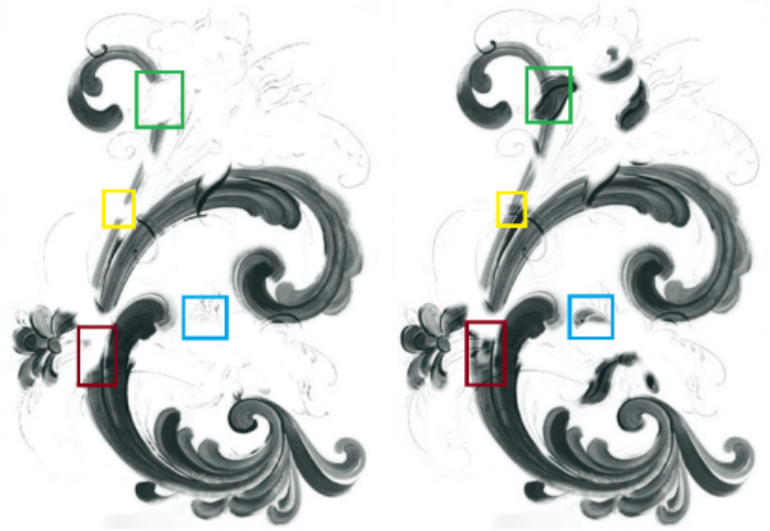


Figure 4.5: Comparison of layer decomposition

Comparison of layer decomposition before and after modification at the 2nd layers. left and right show the results by using $E_{spatial}$ and E_{flow} in Eq 4.1;

Involving ETF filed of Eq 4.2 in $E_{spatial}$, the smoothness penalty is rewritten as,

$$E_{flow} = \frac{1}{n}\sum_{i=1}^{n} \|t^{new}\| \left(\bigtriangledown_{\theta}\alpha_i\right)^2 \tag{4.3}$$

where θ denotes the direction of t^{new}, and $\bigtriangledown_{\theta}\alpha_i$ is the gradient of opacity in the direction of t^{new}. Moreover, we weight this penalty by the norm of t^{new}. Applying the updated E_{flow} to the layer decomposition of Eq 4.1 instead of $E_{spatial}$, the strokes become complete and smooth, which can be noted in Figure 4.5.

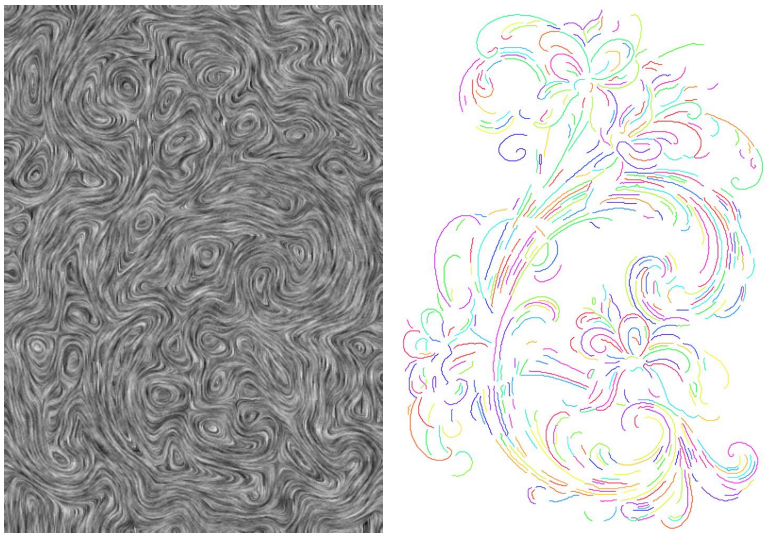


Figure 4.6: Edge Tangent Flow field and coherent lines of a Rosemaling painting. It contains lots of C and S strokes.

Second, the coherent lines as the constraint of brush stroke edges are involved in layer

decomposition of Eq 4.1. Herein, the coherent lines can be computed as follows.

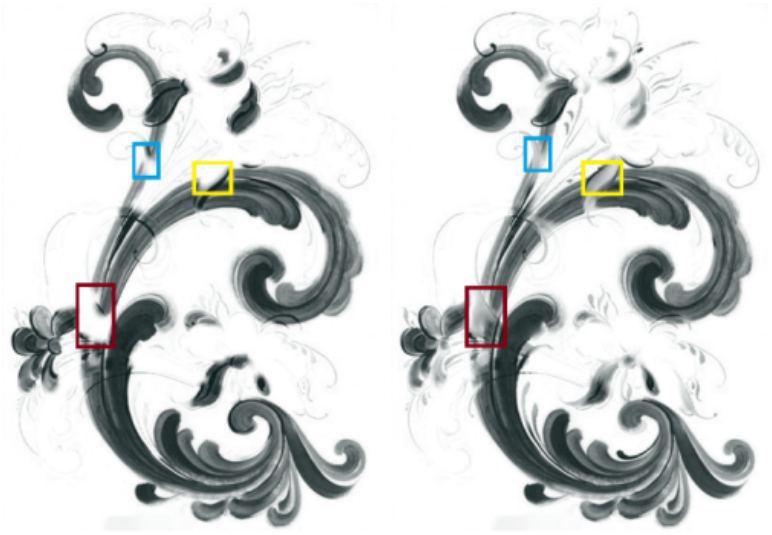
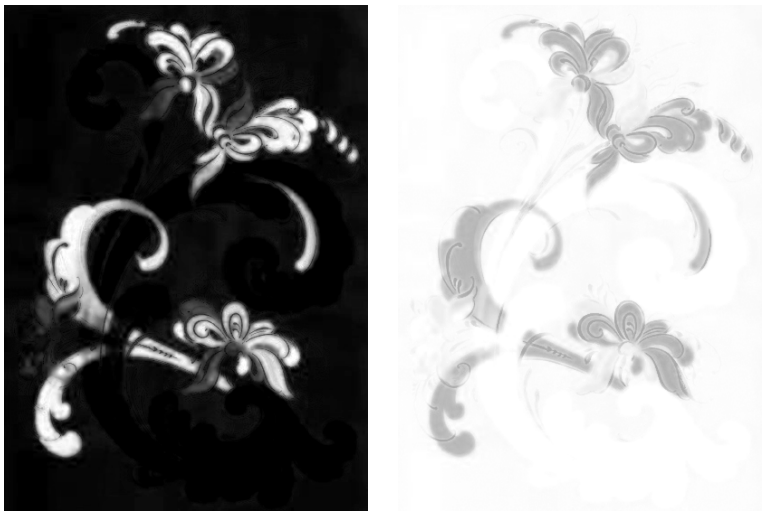


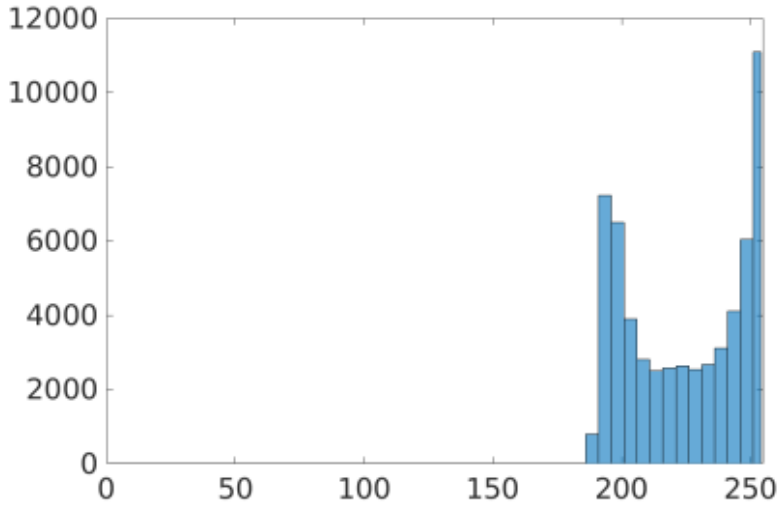
Figure 4.7: Comparison of layer decomposition before and after using E_{egde} in Eq 4.1;

Given a ETF field $t(x)$, the flow-guided anisotropic Difference of Gaussian (DoG) filter is employed, in which the kernel shape is defined by the local flow encoded in ETF field. Note that $t(x)$ represents the local edge direction. It is most likely to make the highest contrast in the perpendicular direction, that is, the gradient direction. When moving along the edge flow, the DoG filter is applied in the gradient direction. As a result, we can 'exaggerate' the filter output along genuine edges, while 'attenuating' the output from spurious edges. This not only enhances the coherence of the edges, but also suppresses noises. Iteratively applying this flow-based DoG filter results in a binary output which reaches a satisfactory level of line connectivity and illustration quality. The coherent lines can be regarded as the edges of brush strokes.

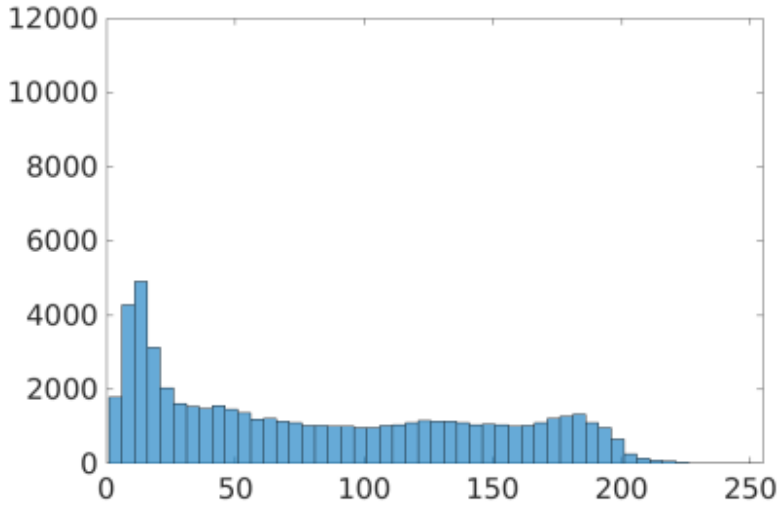


(a) alpha map of layer1 (b) intensity map of layer1

To preserve the stroke edges, we assume that the opacity along the coherent lines is consistent, i.e. $min \int_l \| \bigtriangledown \alpha \|^2 dx$, where l denotes the collection of coherent lines. Hence,



(a) histogram of intensity map of layer1



(b) histogram of intensity map of layer1

Figure 4.9: Comparison between transparency and intensity of layer1

the constraint term is defined by applying Laplacian operator to the opacity along the coherent lines,

$$E_{edge} = \|LY\|^2 \tag{4.4}$$

where all the opacity α_i are stacked in the vector Y, and L denotes the connection matrix. The eight-connected neighboring rule is utilized to construct the connection matrix L, that is, if two adjacent pixels, i and j, stay on the same coherent line, the item of $L(i, j)$ is set to -1 ; otherwise 0. Figure 4.7a and 4.7b shows that the edges of strokes become visible and complete after involving E_{edge} into Eq 4.1. Accordingly, the layer decomposition of Eq 4.1 is rewritten as,

$$E = \omega_{polynomial} E_{polynomial} + \omega_{opaque} E_{opaque} + \omega_{flow} E_{flow} + \omega_{edge} E_{edge} \tag{4.5}$$

where $\omega = 100, \omega = 20$ for all our examples. For comparison, we perform the schemes of Eq 4.1 and Eq 4.5 separately on the same set of brush paintings and compare the root-mean-square-error (RMSE) of the opacity of the coherent lines on each layer shown in Table 1. The RMSE by Eq 4.5 is noticeably less than that by Eq 4.1. This means that the coherent lines have been embedded into the opacity of each layer. The weights are empirically determined in terms of the opacity RMSE of coherent lines.

Table 1 Opacity of Coherent Lines on Layers

Layer#		RMSE by Eq.1	RMSE by Eq.5
Rosemaling1	2	29.12	15.39
	3	12.89	5.92
	4	16.74	9.63
Rosemaling2	2	23.34	11.25
	3	26.33	15.97
	4	14.22	8.26
Chinese1	2	44.41	26.33
	3	25.28	16.37
	4	9.26	4.19
Chinese2	2	17.55	10.32
	3	23.27	11.84
	4	31.94	18.35

Opacity RMSE of coherent lines on each layer is computed by taking the square root of the mean of the variance of every line segment.

Chapter 5

EXTRACTION OF BRUSH STROKES

Brush strokes normally follow certain rules in a brush painting and they vary depending the type of paintings. Here we discuss how we make use of such rules. Rosemaling paintings usually use subtle and vibrant colors to enhance color contrast between overlapped strokes. As a result, the overlapped strokes tend to be classified into different layers. Extracting brush strokes within one layer is easier than directly from the input painting. We employ the Maximally Stable Extremal Regions (MSERs) proposed in [23] and [56] to extract brush strokes since it is invariant to affine intensity changes. MSERs algorithm requires a distinct difference between background and foreground while allowing a small variation of intensity within the selected stroke region. Usually, the strokes on the decomposed layers satisfy this requirement.

However, it is likely that MSERs may fail in segmentation with the following scenarios,
(1) two adjacent regions with the similar intensity;
(2) the region with a high transparency.
Moreover, like the other existing segmentation approaches, the MSERs algorithm encounters over-segmentation issue as well. To tackle these challenges, the coherent lines [54] is introduced into MSERs, which both enhances the edges of strokes and preserves the completeness of strokes. For completeness sake, we briefly address MSERs algorithm and then address our modification.

5.1 MSERs Algorithm

Maximally Stable Extremal Regions (MSERs) [22] can denote a set of distinguished regions that are detected in a gray scale image. All of these regions are defined by an extremal property of the intensity function in the region and on its outer boundary. MSERs have properties that form their superior performance as stable local detector. The set of MSERs is closed under continuous geometric transformations and is invariant to affine intensity changes. Furthermore MSERs are detected at different scales.

This paper introduces MSER tracking, which requires a data structure that can be efficiently built and managed. The component tree is a structure which allows the detection of MSERs within an image and, in addition, constitutes the basis for MSER tracking. The component tree has been recently used for efficient implementation of watershed

segmentation [57] .

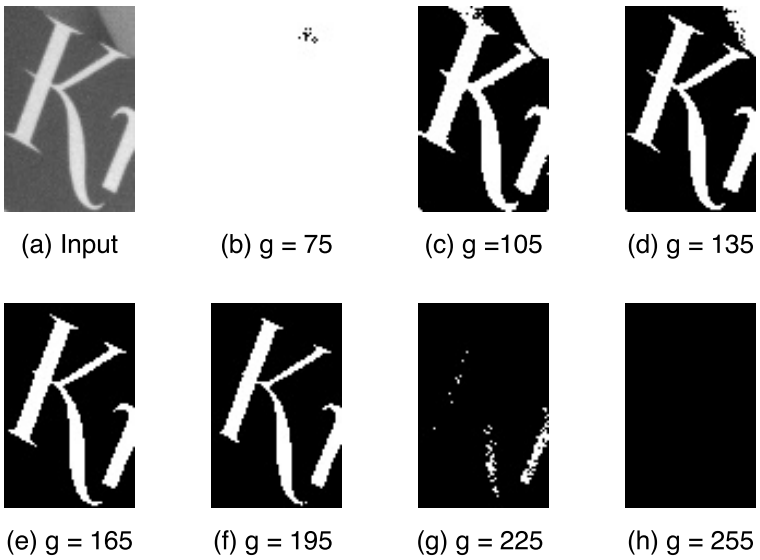


Figure 5.1: Extreme regions based on threshold

Threshold images analyzed during creation of component tree. Figure (a) shows the considered area and figures (b) to (g) the results of thresholding this image at gray level g. The letter k is identified as MSER because the size of the connected region does not change significantly in the gray level range from 135 to 195.

The edges within the tree define an inclusion relationship between the connected regions. Thus, for a region that is the son of another region within the tree, is fulfilled. By moving in the component tree upwards, the corresponding intensity value g of the extremal regions becomes lower, which leads to increased region sizes. The root of the tree represents a region which includes all pixels of the input image . Figure 5.2 shows typical parts of the component tree created for the image shown in Figure 5.1(a).

MSERs can denote a set of distinguished regions that are detected in an intensity image. All of these regions are defined by an extremal property of the intensity function in the region and on its outer boundary, i.e. for a given extremal region S, the internal intensity is more than the intensity of boundary of S,

$$\forall p \in S, \forall q \in \partial S, \longrightarrow I(p) \geq I(q)$$

where ∂S denotes the boundary of S. Changing threshold, the extremal regions may further split or merge. The resulting extremal regions may be represented by the component tree. Accordingly, we may compute the change rate of the area of extremal region by,

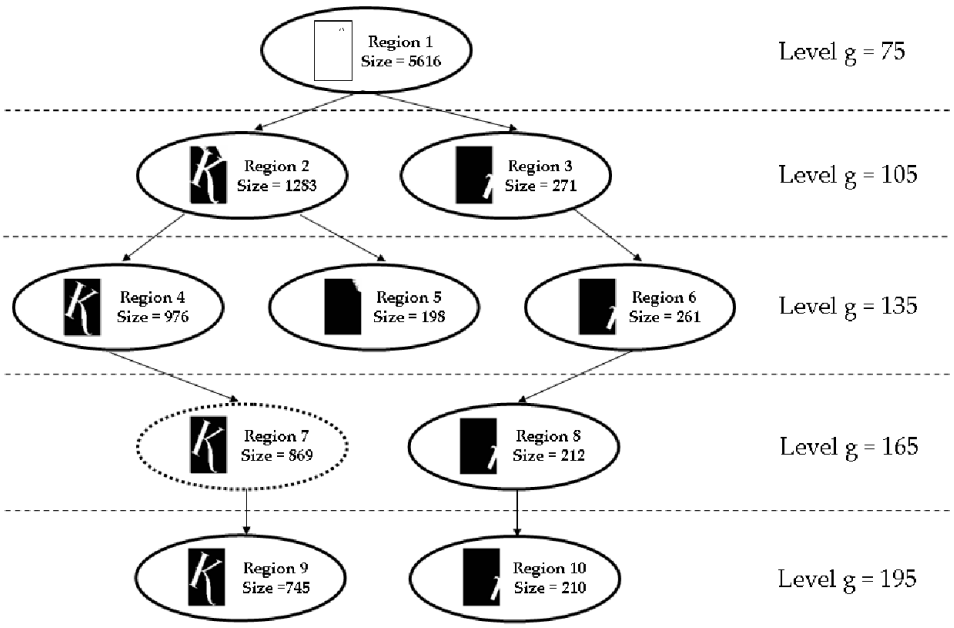


Figure 5.2: MSER tree

Parts of the created component tree for the input image. Region 7 is identified as MSER.

$$\gamma(S_i^g) = \frac{\left(\left|S_j^{g-\Delta}\right| - \left|S_k^{g+\Delta}\right|\right)}{|S_i^g|}$$

where $|.|$ denotes the cardinality, S_i^g is the i-th region which is obtained by thresholding at an intensity value g and Δ is a stability range parameter. $g-\Delta$ and $g+\Delta$ are obtained by moving upward and downward respectively in the component tree from the region S_i until a region with intensity value or is found. $\{i, j, k\}$ are the indices of nodes of the component tree. MSERs correspond to those nodes of the component tree that have a stability value γ, which is a local minimum along the path to the root of the tree.

Figure 5.3: MSERs on the intensity of image

Perform the original MSERs on the intensity of image; It is clear to see over-segmentation.

5.2 Modified MSERs Algorithm

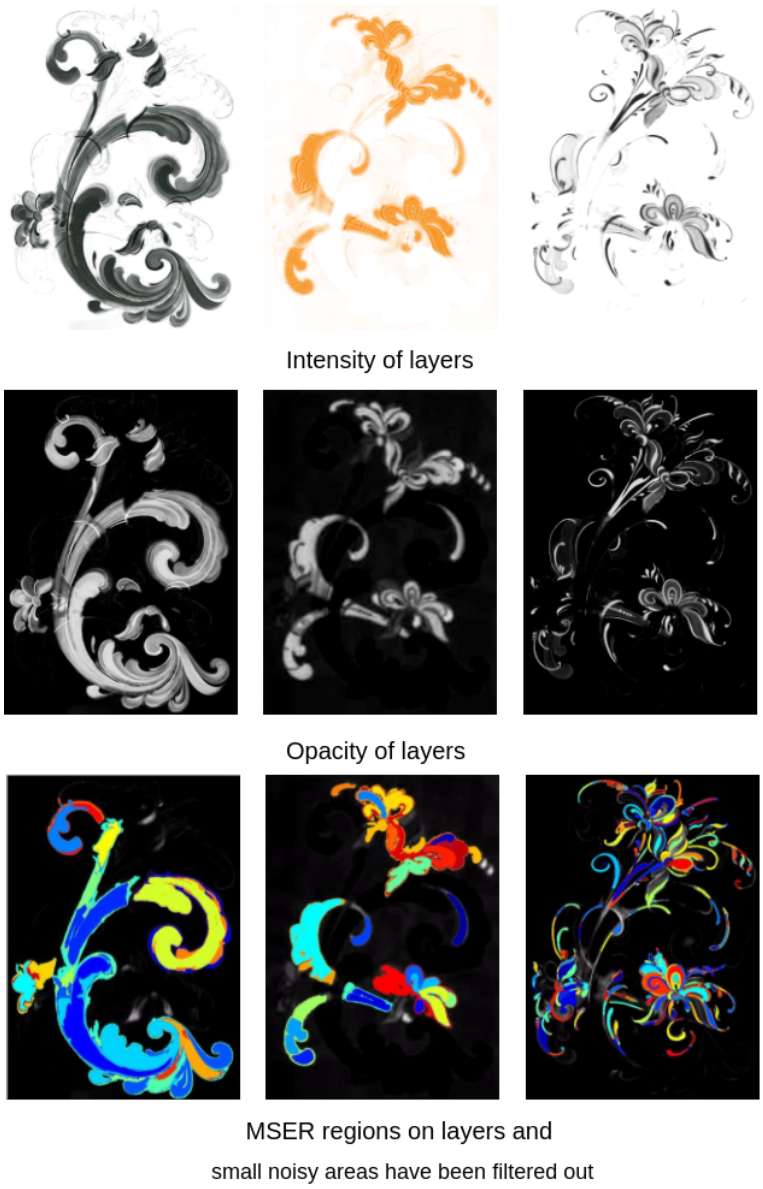


Figure 5.4: The intensity, opacity maps, and MSER regions of three layers by the modified MSERs.

In terms of the definition of the area change rate γ, MSERs may fail in segmentation with the following scenarios,
(1) the region with the intensity very close to the background;
(2) two adjacent regions with the similar intensity;
(3) the region with a very high transparency.
The opacity of the image, α, is always independent of the intensity of the image. It is likely to avoid the abovementioned scenarios if segmentation can be performed on γ. We perform the layer decomposition of Eq 4.5 on a brush painting and show the intensity and opacity of one layer associated with the individual histograms in Figure 4.9. It can

be noted that the opacity of the layer contains richer layered details than the intensity. Moreover, Table 1 also shows that the opacity of the layers is more suitable for stroke segmentation than the intensity. The first modification is to perform MSERs on the opacity of every layer. We aim at the scenarios of two adjacent regions with the similar opacity. When the extremal region is growing up through changing threshold, it is feasible to restrict the region by introducing the coherent lines. According to the definition of the extremal region, the boundary of region S should satisfy,

$$\forall p \in S, \forall z \in \bar{S}, \forall q \in \partial S \longrightarrow I(p) \geq I(q) \text{ and } I(q) \leq I(z)$$

where $\bar{S}$ denotes the complement of S. The second modification is to simply modify the opacity of layers, that is, overlapping the coherent lines with the layer and then changing the opacity of coherent lines to the smallest value in the layer.

To deal with the over-segmentation issue, the coherent lines play an important role. Given a region S, we modify the area change rate γ as,

$$\gamma(S_i^g) = \frac{||S_j^{g-\Delta}| - |S_k^{g+\Delta}||}{|S_i^g|} + \frac{||Q_j^{g-\Delta}| - |Q_k^{g+\Delta}||}{|Q_i^g|} - (1 - \frac{|Q_i|}{|\partial S_i^g|}) \tag{5.1}$$

where Q denotes the set of pixels which stay on the coherent lines and $Q \subset \partial S$. The third modification is to take into account the change of coherent lines to the boundary of S, i.e. the third term penalizes that a small portion of the boundary ∂S is occupied by coherent lines.

Figure 5.4 shows the segmentation results by the modified MSERs, which correspond to brush strokes. It can be noted that performing MSERs on the intensity of image inevitable yields over-segmentations. Performing the modified MSERs on the opacity of layers, the strokes tend to complete and smooth within one layer. Moreover, some small regions with the distinct opacity values against neighboring areas have been filtered out.

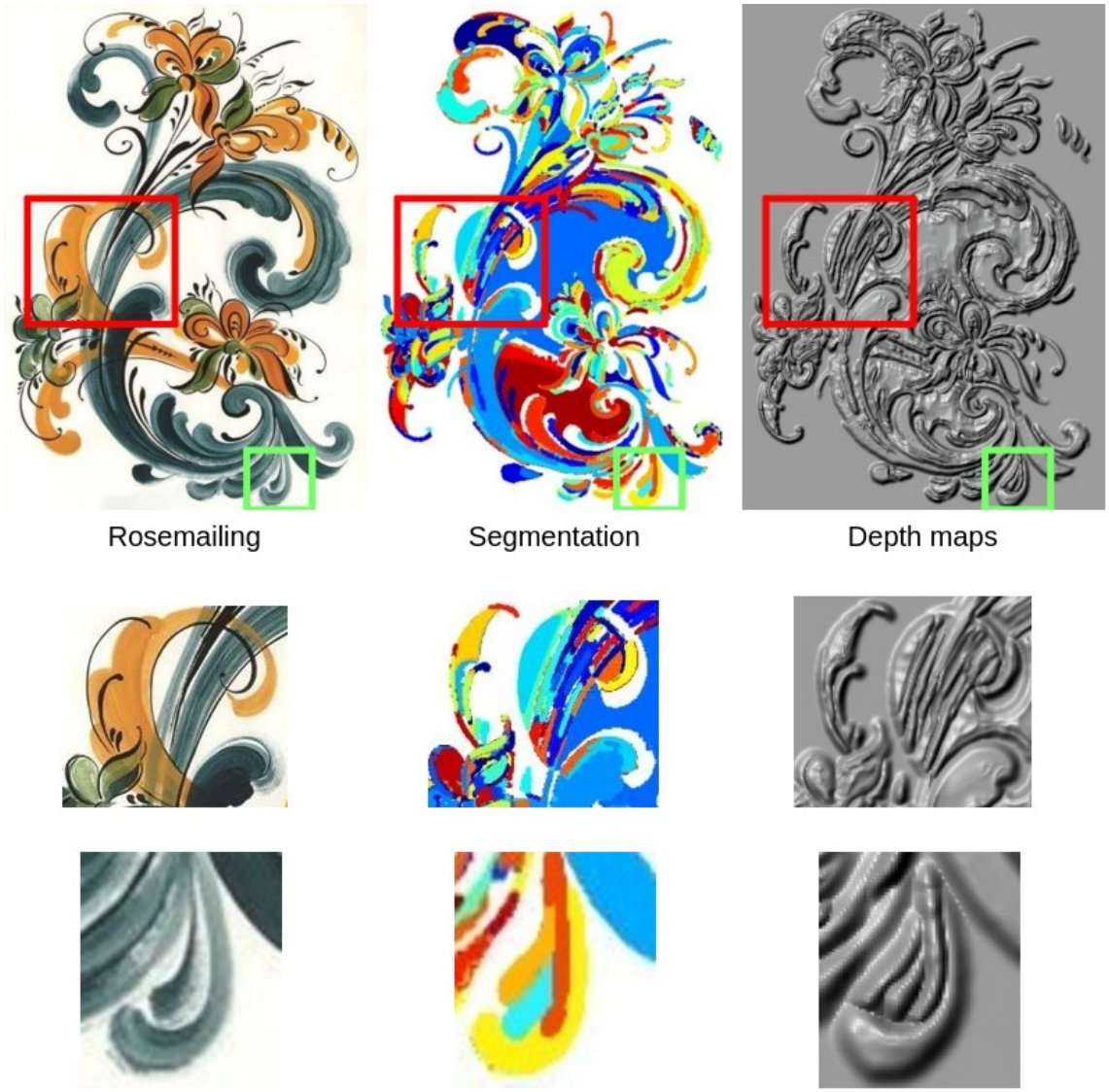


Figure 5.5: Incorrect and over segmentation

Resulting depth maps based on segmentation of image intensity. Second row shows the incorrect segmentation (brown area in Top is missed in Middle);Last shows the over-segmentation (2 patches in Top are divided into 4 parts in Middle).

Chapter 6

DEPTH MAPS AND "3D STROKES"

Since the brush strokes can be extracted individually, it is natural to independently generate the individual depth maps and then merge the depth maps together to form the desired bas-relief. The image-based bas-relief approaches[4] [16] perform SFS separately on the segmented regions of an image. Most of segmentation methods work on image intensity. Due to lack of information of the spatial relationship, they always suffer from the issues of over-segmentation or incorrect segmentation. As a result, post-processing with user assistance to manipulate the depth maps is necessary. To illustrate this issue, we applied the Poisson reconstruction method [4] on the segmented Rosemaling painting shown in Figure 4.3 to generate the depth maps. It can be seen that there are lots of over-segmentation, or even incorrect segmentation as shown in Figure 5.5. The resulting independent depth maps have to be piecewisely merged, rescaled or deleted by users, which is not trivial and tedious. Thus, in our implementation, we employ the orthogonal SFS [27] on the segmented strokes. These extracted strokes take into account the spatial occlusion and are suitable for 3D manipulation. However, the SFS is performed on the opacity of the image rather than the intensity. The brightness equation used in SFS is expressed as,

$$I(x) = \frac{1}{\sqrt{1 + |\bigtriangledown h|^2}}$$

It can be noted that the higher the intensity I, the smaller the change of depth h. Usually, some brush strokes tend to produce a high intensity in a painting. As a result, if the SFS is performed on intensity, the resulting stroke models will become flat and lack of hierarchy. Moreover, for a region, the boundary height is noticeably higher than its inside. This is because the gradient of the edges is always more than that of the inside. The opacity of image is independent of the intensity (see Figure 4.9). Each stroke has an appropriate distribution of opacity, which is in favor of a layered look.

To deal with the issue of unbalanced growth, we rewrite the Eikonal Equation as,

$$\|\bigtriangledown h\| = \sqrt{\frac{1}{\|I(x)\|^2} - 1 + \Delta} \tag{6.1}$$

where Δ is a positive displacement. This modification may make the surface inflated.

Moreover, it is desirable to allow the users to recompose the brush stroke models in 3D space for secondary creation purposes. To highlight the term of "3D strokes", we present three approaches based on the generated "3D stroke" models here.

Rising a Stroke

For a set of stroke models, we may quantify their depth maps within a specified height scope. The order of strokes may be given by users. Suppose there are n overlapped strokes in a painting, which are sorted in a descending order from 1 to n. Let S_i be the i-th stroke and $h_i(x)$ for the depth of point x within S_i . The depth map of S_i is quantified within the scope of $[0, \rho]$ as,

$$h_i(x) = \frac{(\rho - (i-1)d)h_i(x)}{H_i} \tag{6.2}$$

where $H_i = \frac{1}{|S_i|} \sum_{x \in S_i} h_i(x)$, and d controls the depth difference between the successive order of stroke models. To manipulate stroke models, such as raising a stroke, this can be fulfilled by simply changing the order of as shown in Figure 6.1. However, the scheme of Eq 6.2 cannot settle the issue of two strokes obstructing each other. For example, stroke A partially obstructs the other stroke B while it may be partially obstructed by B. This can be resolved by the following stitching approach.

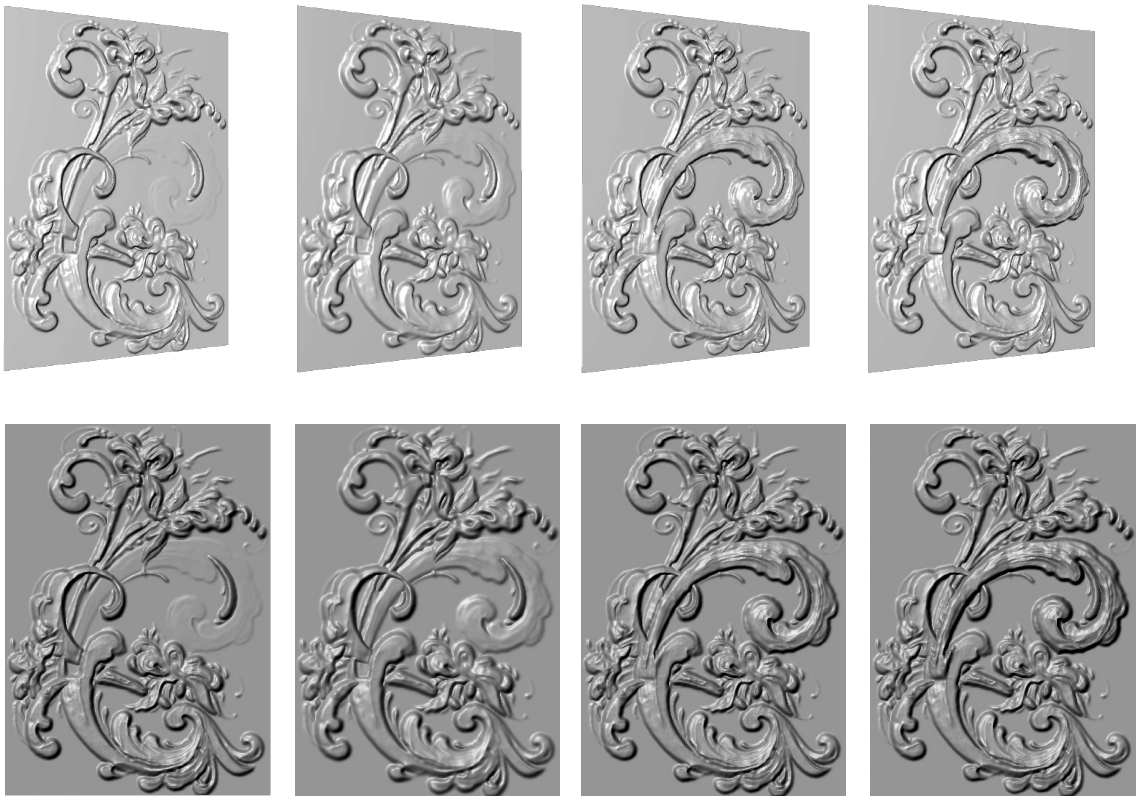


Figure 6.1: Rising a stroke

The selected stroke is rising from bottom to top in layer order.
There are 4 layers.

Stitching one Stroke on Another one

For two overlapping strokes S_i, S_j, it is desired to stitch the model of S_i on that of the other S_j instead of overlapping their depth maps. This may be accomplished by solving for depth functions $h_i : S_i \rightarrow R$ such that the resulting h_i satisfies stitching positional constraints. We define the set of $D_{ij} \subseteq \partial S_i \cap \partial S_i$ as the stitching constraints. The depth functions are solved by minimizing the Dirichlet energy,

$$\int_{S_i} \| \bigtriangledown h_i - \vec{v_i} \|^2 dx, \;\; s.t. \; h_i(x) = h_j(x), \forall x \in D_{ij} \tag{6.3}$$

where $\vec{v_i}$ denotes the differential coordinates of the model of S_i. Intuitively, this is to update $h_i(x)$ which keeps close to the depth map of S_i as much as possible while satisfying the positional constraints in a least squares sense. We perform the scheme of Eq 6.3 on the overlapping area of multiple strokes and illustrate all the possible spatial occlusion cases in Figure 6.3. It can be noted that there are four overlapped strokes to be depicted by using transparency in the Rosemaling painting. In fact there are at most eight occlusion cases as shown in Figure 6.3. We can further change the shape of these 3D strokes if needed.

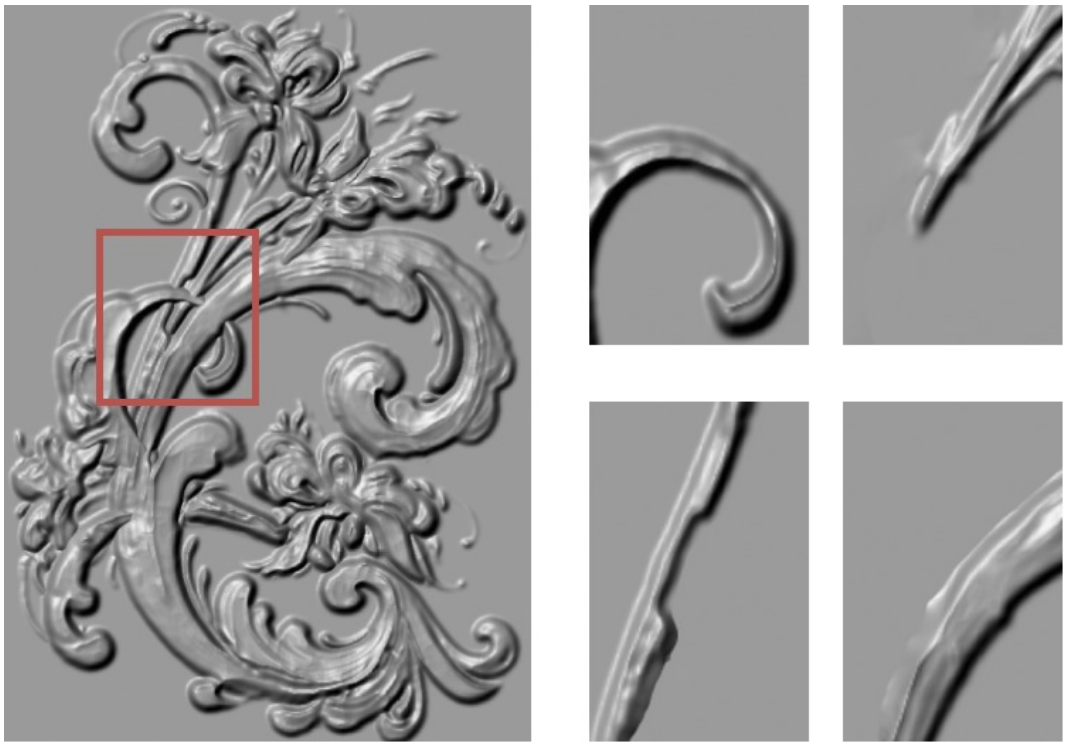


Figure 6.2: Strokes segmented in selected region

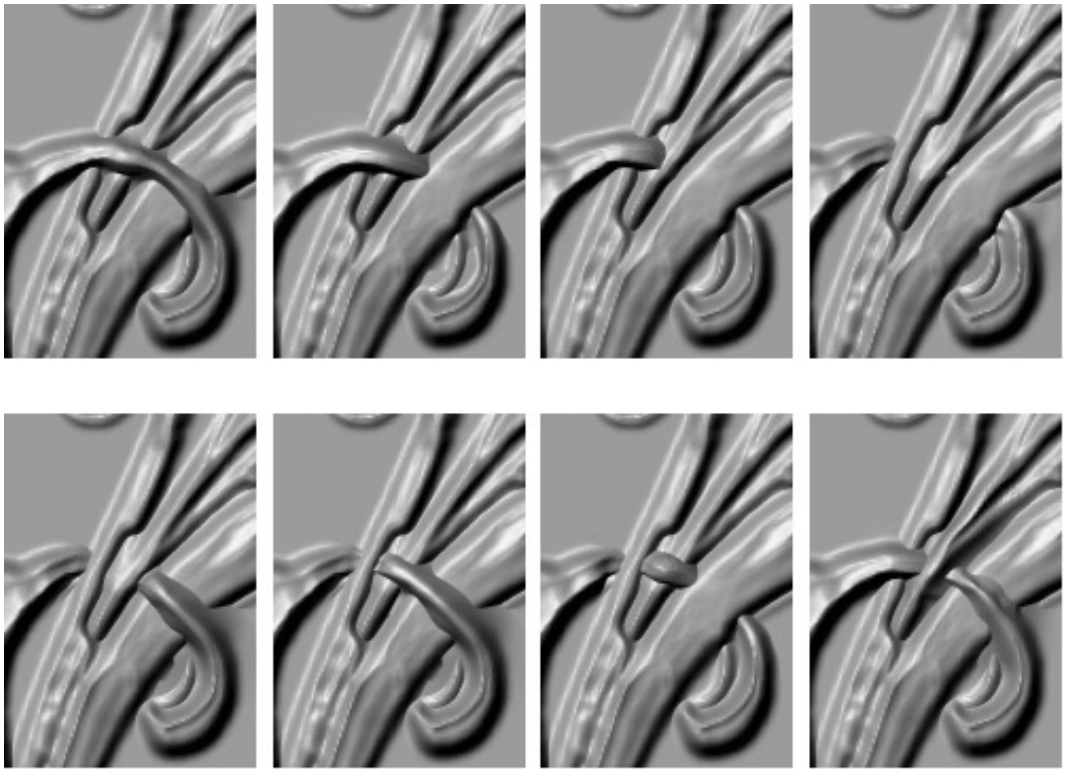


Figure 6.3: The result of stitching strokes

The result of stitching one stroke on another one.
8 cases are highlighted by combining 4 selected strokes

Changing the Shape of Strokes

The key point is to determine the changed boundary of strokes on the background plane at first. For a given stroke S_i, the boundary of S_i on the plane corresponds to the positional constraints in Eq 6.3. When users change the boundary ∂S_i, the shape of the model of S_i will be changed accordingly based on Eq 6.3. Painters are used to sketching a skeleton $D_i^{'}$ on the background plane. It requires us to change the boundary of S_i on the plane according to $D_i^{'}$.

To this end, S_i is triangularized by a regular grid on the plane, and its skeleton may be extracted by the Hilditch thinning method [58]. This skeleton can be matched to $D_i^{'}$ by the lengths. The shape of boundary is deformed on the plane in terms of $D_i^{'}$. For an As-Rigid-As-Possible shape deformation [59] on a plane, we employ the curve Laplacian coordinates on the boundary of S_i, $\|LV - \delta(V)\|^2$, where V denotes the vertices of boundary ∂S_i; the mean value coordinates $\|MV\|^2$, where V denotes the vertices within S_i; and the positional constraints $D_i^{'}$, $\|CV - V^{'}\|^2$, where $V^{'} \in D_i^{'}$ while V for the updated vertices. The boundary of S_i is updated on the plane by minimizing the following sum of these three terms,

$$\|LV - \delta(V)\|^2 + \|MV\|^2 + \|CV - V^{'}\|^2 \tag{6.4}$$

Herein, we expand the matrices of L, M, C through adding zero elements, such that V contains all the vertices of S_i in Eq 6.4. The resulting boundary of S_i is then utilized in

Eq 6.3 to update the depth functions of S_i . Figure 6.4 shows how to change the shape of the model of S_i.

(a) selected stroke (b) modified depth map

Figure 6.4: The shape of a stroke is changed

Remark Compared with image-based bas-relief generation, the main advantage of our approach is the introduction of "3D strokes". Spatial occlusion offers an important cue in conveying the spatial relationship of the depicted objects and the depth information. Unlike the usual image segmentation problems, our method takes into account spatial occlusion in our stroke segmentation. As a result, over-segmentation and incorrect segmentation are largely eliminated. Moreover, the extracted "3D strokes" both help us sort out the stroke order and provide us a chance to rearrange the strokes as needed. This is in favor of further artistic creation in bas-relief design.

Chapter 7

RESULTS

We use the published codes of the layer decomposition and MSERs, which are available on GitHub (at: https://github.com/CraGL/Decompose-Single-Image-Into-Layers; and https://github.com/idiap/mser), and performed the proposed approach on eight brush paintings, including Rosemaling and Chinese brush paintings. All the paintings are from the internet. Compared to Rosemaling, Chinese brush paintings do not emphasize the use of vibrant colors, even the adjacent strokes may share the same color in some paintings. The boundary of the overlapped brush strokes may be very blurred. Thus, we select some of the Chinese brush paintings for a test in terms of the boundary of strokes and the use of transparency. All the tests were performed on a 6-core of 3.33 GHz Intel Core Xeon CPU with memory of 32 GB(RAM). In our implementation, the parameters in Layer decomposition, ETF field, and MSERs are set the default values as in the original codes. Table 2 further shows the running time of the proposed approach. Compared to the performance in [20] and [56], there is no distinct difference. Additionally, SFS does not consume much time since it depends on the segmented stroke regions. Our implementation is not multithreaded. All the resulting images are available in the next section.

Table 2 Performance

No.	Stroke number	Runtime of Layer (sec)	Runtime of MSERs(sec	Runtime of SFS (sec)
Rosemaling1 (548*732)	102	371.07	0.62	25.58
Rosemaling2 (564*858)	55	402.11	0.55	12.54
Rosemaling3 (472*784)	82	324.72	0.84	21.71
Rosemaling4 (357*651)	126	218.6	0.78	34.21
Rosemaling5 (556*668)	241	173.83	0.4	47.68
Rosemaling6 (450*349)	63	212.77	0.31	14.26
Chinese1 (546*725)	38	71.51	0.22	10.85
Chinese2 (594*725)	213	574.98	1.38	43.22

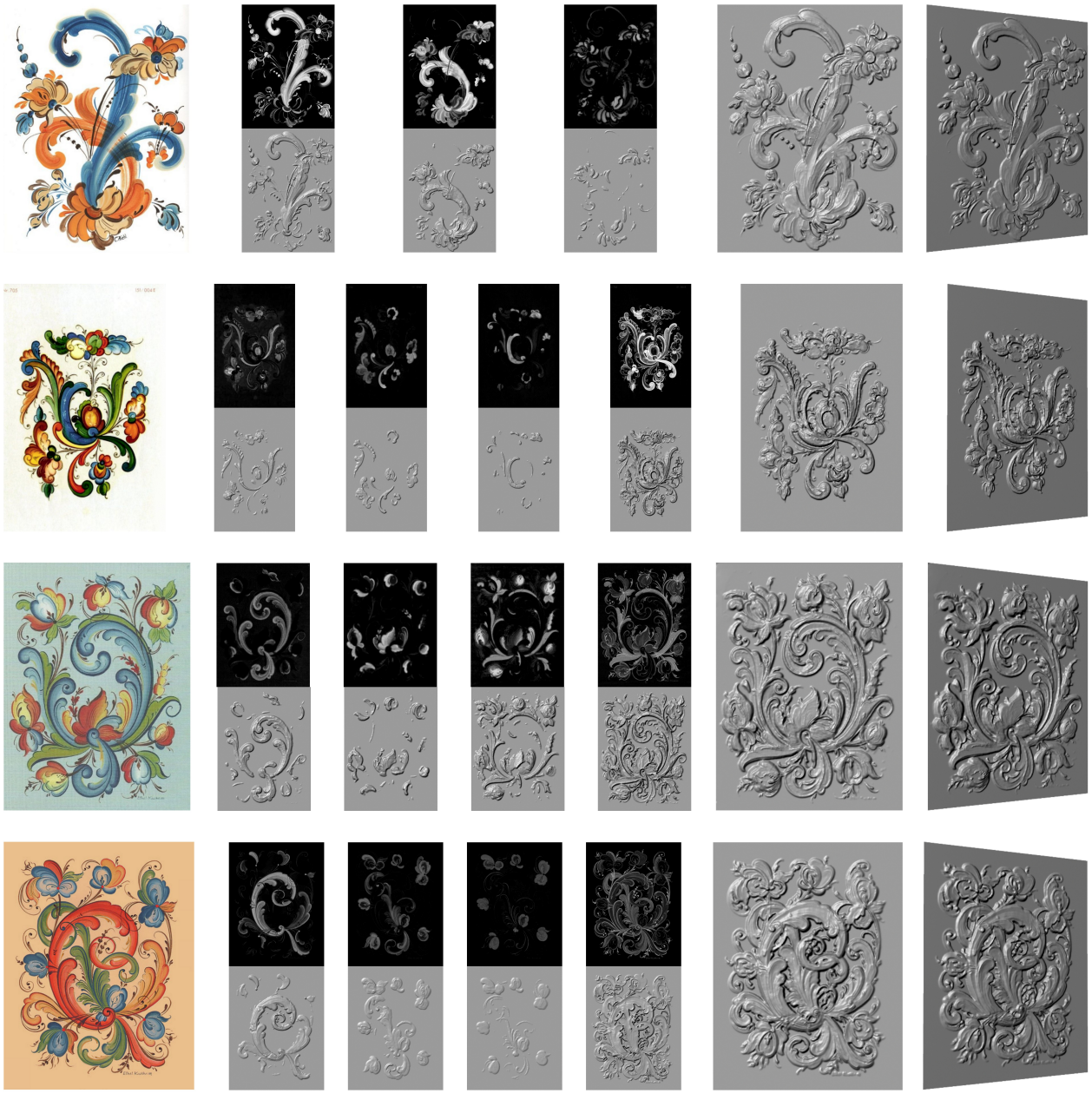
7.1 More Results

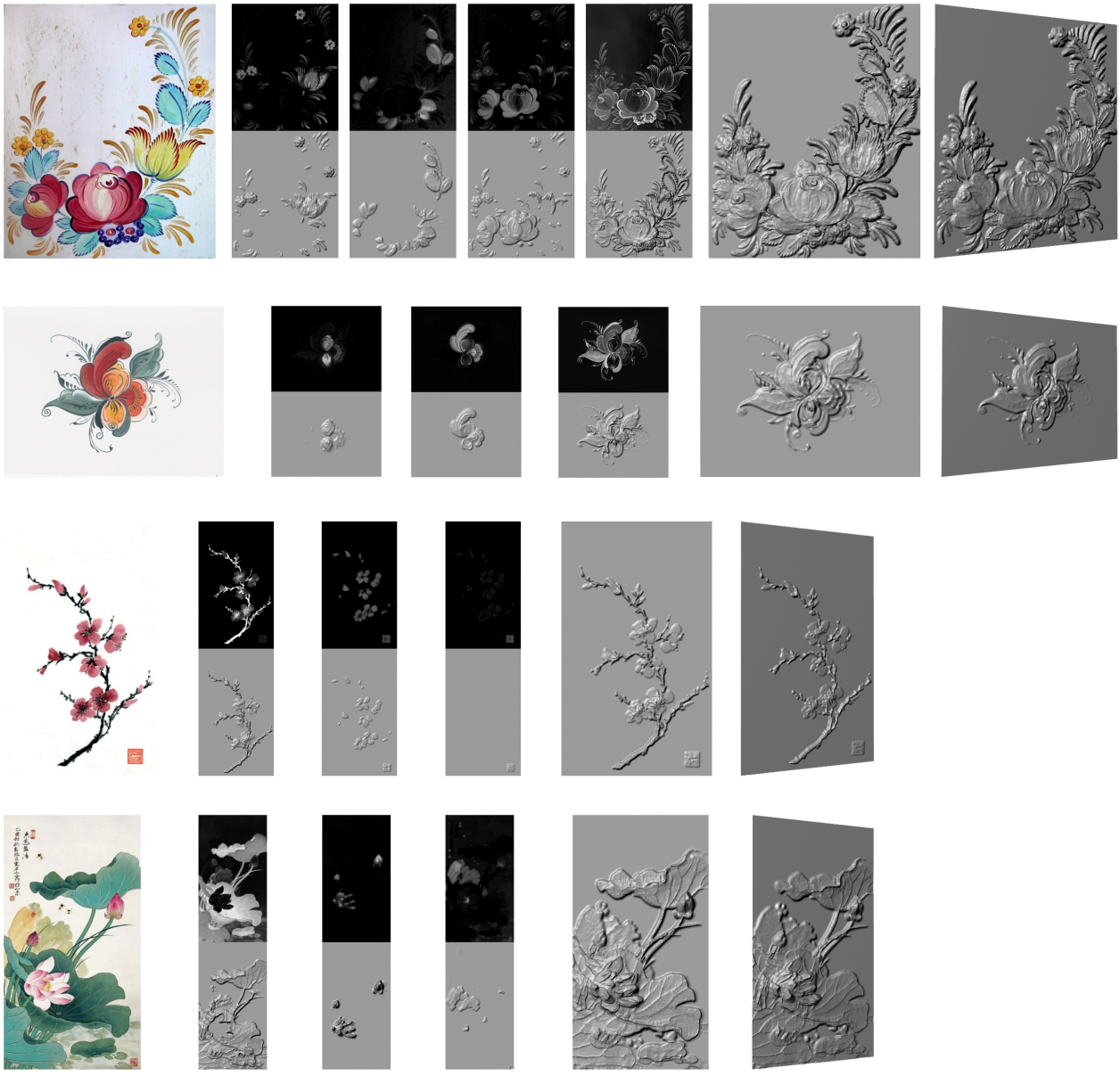
More bas-relief generation based on different brush paintings

Table S1 Performance

No.	Layer#	Opacity RMSE of coherent lines		Runtime of Layer (sec)	Runtime of MSERs (sec)	Runtime of SFS (sec)
		By Eq.1	By Eq.5			
Rosemaling 1 (548*732)	2	29.12	15.39			
	3	12.89	5.92			
	4	16.74	9.63	371.07	0.62	25.58
Rosemaling 2 (564*858)	2	23.34	11.25			
	3	26.33	15.97			
	4	14.22	8.26	402.11	0.55	12.54
Rosemaling 3 (546*725)	2	17.58	14.45			
	3	19.78	12.77			
	4	24.63	17.24	324.72	0.84	21.71
Rosemaling 4 (400*522)	2	34.11	21.42			
	3	20.52	14.58			
	4	22.2	17.99			
	5	24.24	11.51	218.6	0.78	34.21
Rosemaling 5 (556*668)	2	19.44	12.84			
	3	31.22	24.71			
	4	30.72	19.95			
	5	38.98	21.36	173.83	0.4	47.68
Rosemaling 6 (450*349)	2	30.48	18.75			
	3	26.47	12.74			
	4	26.86	18.41			
	5	17.85	8.65	212.77	0.31	14.26
Chinese 1 (472*784)	2	44.41	26.33			
	3	25.28	16.37			
	4	9.26	4.19	71.51	0.22	10.85
Chinese 2 (357*651)	2	17.55	10.32			
	3	23.27	11.84			
	4	31.94	18.35	574.98	1.38	43.22

The stroke segmentation and modeling are based on the opacity of layers in our implementation. Thus, we show the opacity RMSE of coherent lines in order to emphasize that the coherent lines have been embedded into the opacity of layers.





Chapter 8

CONCLUSIONS

Bas-relief may be generated from both 3D models and 2D paintings. Our approach can effectively extract brush strokes from 2D paintings and generate the individual depth maps for bas-relief composition in 3D. Since the depth maps of brush strokes are independent of each other, the order and shape may be redefined by the users. It is suitable for Recomposition in bas-relief designs. Limitations In our implementation, we employ layer decomposition in order to split the overlapped brush strokes into different layers, so that the strokes can be easily segmented within one layer. However, the main challenge is the overlapped strokes may appear on the same layer. Rosemaling painting designs use subtle and vibrant colors to enhance the color contrast between adjacent brush strokes. Moreover, for a layered look, transparency is applied and is variable with a big scope. Thus, layer decomposition benefits the overlapped brush strokes separated on Rosemaling paintings. However, Chinese brush paintings do not emphasize the use of vibrant colors, even the adjacent strokes may share the same color in some paintings. The boundary of the overlapped brush strokes may be very blurred. This may result in stroke segmentation failure.

Chapter 9

Future Work

In the future, At first,we will study a wider range of paintings and investigate the issues of layer decomposition and stroke segmentation. Different paintings have different stroke patterns. Understanding the rules will improve the success rate of a wider range of paintings. Proper automatic color separation in the overlap regions is not trivial which will also be studied in the future.Second we would try to build high relief based on current algorithm, and based on the depth of high relief , we would try to generate 3D brush painting effects from 2D brush painting, as showed in Figure 9.1.

Figure 9.1: 3D brush painting : Autumn Tree

The top row shows two different views of a painted tree. The bottom row shows the respective proxy geometry with paint stroke paths on top.

9.1 From 2D Brush Painting to 3D Brush Painting

9.1.1 Abstract

With the increasing popularity and development of virtual reality, digital painting on 2D canvases is now being extended to 3D spaces. 3D painting as a new art form are now widely accepted among artists. Software on VR platform such as Tilt Brush and Oculus Quill make 3D painting more and more intuitive for 2D painters. We aim to build a system which enable artist to transfer a given 2D brush painting to 3D painting, in a way that gives creative freedom to the artist while maintaining an acceptable level of controllability based on the given 2D brush painting.

We address this problem into four main steps: first, segment strokes from the given 2D brush painting. Since in 3D brush painting, strokes are placed in 3D space based on user input, to mimic such effect, successful strokes segmentation is quite essential. Brush strokes may have high diversity of colour, so layer decomposition may need to be applied. Second, stroke refinement, since brush strokes may be over segmented or under segmented, stroke need to be refined. A refinement method based on user input need to be applied. Third, transfer 2D strokes to 3D strokes, simulate the effect the 3D stroke with supplied 2D stroke, and 3D volumetric painting would be applied [60]. Fourth, 3D strokes placement, based on the layer order and user inputs, we place and blend 3D strokes in 3D space.

9.1.2 Aims and Objectives

With given 2D brush painting, we aim to design a new approach to generate a 3D brush painting. With such a goal, we need to achieve following: 1. Based on edge and region segmentation, retrieve the reasonable palette colours from input 2D brush painting. 2. Efficient hierarchical segmentation for brush strokes based on layer decomposition. Simulate different scale of brush strokes based on hierarchical segmentation and dynamic branch cutting. 3. Refine stroke segmentation based on user input. 4. Generate correspond 3D brush strokes from 2D brush strokes. 5. Place and deform 3D brush strokes based on user input. 6. The 3D brush painting must preserve the style and placement order of original 2D brush.

9.1.3 Background and Literature Review:

Stroke based rendering:

The difference between primary space (the 3D world in which objects live) and secondary space (the 2D canvas on which depictions of those objects are created) was highlighted by [61].

The field of non-photorealistic rendering (NPR) has developed many stylization on 2D canvas. Although traditional photorealistic rendering research focuses on the primary space (e.g., scene representation, visibility determination, global illumination), the NPR community first approached the problem from the opposite direction by focusing entirely on the secondary space of the 2D canvas [62].

Haeberli's interactive "Paint By Numbers" system [63] started the research of stroke based rendering, which proposed a method to fills a 2D canvas with brush strokes when parametric brush stroke are controlled based on given photograph. And similar works which provide automatic approach by placing discrete elements such as paint strokes or stipples to create nonphotorealistic imagery are described stroke-based rendering [64] . Interactive stylization of images, video, and animations [65] has also been done. These researches mainly focus on how to implement different expressive styles on canvas which algorithmic mapped from primary space.

3D painting system:

The concept of 3D painting system was introduced for more than two decades ago. Hanrahan [62] firstly present a 3D painting system that capable of painting on 3D models, but their work didn't put transparency and fine detailed painting in consideration.

From works of Daily et al. [66], 3D painting applied texture map for painting details effect. However, distortion or seams would happen due to 2D parametrization. Meier [1996] first proposed generating brush strokes attached to 3D objects. Deep Canvas [67], firstly projected painted strokes on the object's surface and dynamic 3D camera can be applied which is hard to achieve using traditional 2D paintings. The WYSIWYG NPR system [68] focused on algorithmic rendering techniques, which enable artist to achieve silhouette stylization and control hatching directly via a painting interface. Maya Paint Effects (Paint Effects 2011) also projects painted strokes onto the surface of scene geometry and uses them as seed points to create new geometric primitives such as grass or flowers.

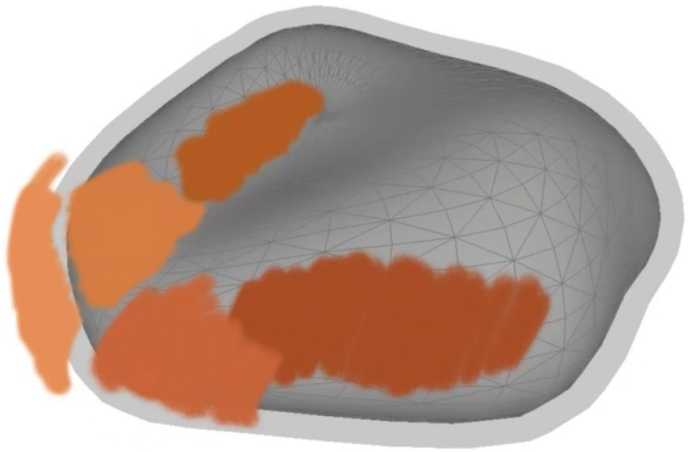


Figure 9.2: 3D brush painting on surface level set

More recently, Overcoat [61] introduced an implicit canvas for 3D painting, which enable creating approximate 3D proxy geometry to shape the implicit canvas, it allows

us to implement tools for painting along level set surfaces or across different level sets. which means we can project parametric strokes onto the underlying 3D models as showed in Figure 9.2. These painting systems are focusing on how to paint on surface of known models. In CanvoX [60] extended painting surface to volumetric painting, with GPU-based Octree [69].

With the rapid development of VR systems, it is natural to enable artists to paint in 3D space. In CavePainting [70] firstly create 3D analogy of 2D brush strokes, to create 3D works in a Cave environment. Softwares like Tilt Brush (1) and Quill (2) have recently surged and are now widely accepted by artists to be positioned as a new art form kim's work [60]. For these painting system, quad strips are rendered as painting strokes, which are fully based on user input.

It is natural to think how to transfer 2D brush paintings to 3D brush painting while preserving its original style.

Outline of Proposed Methodology:

Layer decomposition: Digital painting with different layers is an integral feature of digital image editing software, such as Photoshop and Sketchbook. Layers offer an intuitive way to edit the colour and geometry of components and localize changes to the desired portion of the final image. Without layers, brush stroke segmentation becomes extremely challenging, since they can overlap and blend with each other. So, at first, we decompose the given 2D brush painting into several layers, which layer contains same colour while each pixel has different transparency.

Stroke segmentation: As mentioned above, to transfer 2D strokes to 3D strokes, we need to segment each stroke on the painting, inspired by the level set representation of 3D painting in Overcoat, we would segment brush strokes into a hierarchical structure.

Stroke refinement: To maintain the style of 2D brush painting, successful stroke segmentation is essential for the next step. Wrong segmentation would happen in the process of automatic segmentation, to refine the strokes, user input would be applied. Since the strokes would be a hierarchical structure, hierarchical cut might be applied as well.

3D stroke analogy: Given segmented strokes, we would create correspond 3D strokes, in which 3D volumetric painting would be applied [60].

Stroke placement: After generation of 3D stroke analogy for each 2D stroke, a user interface would be supplied for stroke placement.

Proposed timescale for the work:

Refine the first step of layer decomposition, with the coherent edge of the given 2D brush painting, calculate the palette colour. (4 weeks)

Stroke segmentation into a hierarchical structure, currently, SLIC and hierarchical clustering are in consideration. (4 weeks)

Design the user interface for refine the stroke segmentation. (4 weeks)

3D stroke analogy design. With given segmented strokes, generate corresponding 3D stroke, with detail and style preserved. (6 weeks)

3D stroke placement design. Design the interface for user input, in which 3D strokes can be placed naturally in 3D space while maintaining an acceptable level of controllability.

(3 weeks)

Optimization. (4 weeks)

Table 9.1: Schedule of Future Research Plan

Research Activity	**Target Completion Date**
Design and implement the proposed refinement work. Do experiments accordingly.	01/08/2016
Design the user interface for refine the stroke segmentation.	01/09/2016
3D stroke analogy design and 3D stroke placement design .	01/12/2016
Optimization.	01/01/2017
Write final Thesis for PhD graduation	01/06/2018

Bibliography

[1] Tim Weyrich, Jia Deng, Connelly Barnes, Szymon Rusinkiewicz, and Adam Finkelstein. Digital bas-relief from 3d scenes. In *ACM Transactions on Graphics (TOG)*, volume 26, page 32. ACM, 2007.

[2] Jens Kerber, Art Tevs, Alexander Belyaev, Rhaleb Zayer, and Hans-Peter Seidel. Feature sensitive bas relief generation. In *Shape Modeling and Applications, 2009. SMI 2009. IEEE International Conference on*, pages 148–154. IEEE, 2009.

[3] Jonathan Barron and Jitendra Malik. Color constancy, intrinsic images, and shape estimation. *Computer Vision–ECCV 2012*, pages 57–70, 2012.

[4] Qiong Zeng, Ralph R Martin, Lu Wang, Jonathan A Quinn, Yuhong Sun, and Changhe Tu. Region-based bas-relief generation from a single image. *Graphical Models*, 76(3):140–151, 2014.

[5] Jing Wu, Ralph R Martin, Paul L Rosin, X-F Sun, Frank C Langbein, Y-K Lai, A David Marshall, and Y-H Liu. Making bas-reliefs from photographs of human faces. *Computer-Aided Design*, 45(3):671–682, 2013.

[6] Marc Alexa and Wojciech Matusik. Reliefs as images. *ACM Trans. Graph.*, 29(4):60–1, 2010.

[7] Peter N Belhumeur, David J Kriegman, and Alan L Yuille. The bas-relief ambiguity. *International journal of computer vision*, 35(1):33–44, 1999.

[8] James F Blinn. Simulation of wrinkled surfaces. In *ACM SIGGRAPH computer graphics*, volume 12, pages 286–292. ACM, 1978.

[9] Paolo Cignoni, Claudio Montani, Claudio Rocchini, and Roberto Scopigno. A general method for preserving attribute values on simplified meshes. In *Visualization'98. Proceedings*, pages 59–66. IEEE, 1998.

[10] Songhua Xu, Yingqing Xu, Sing Bing Kang, David H Salesin, Yunhe Pan, and Heung-Yeung Shum. Animating chinese paintings through stroke-based decomposition. *ACM Transactions on Graphics (TOG)*, 25(2):239–267, 2006.

[11] Yu Chen, Tae-Kyun Kim, and Roberto Cipolla. Inferring 3d shapes and deformations from single views. *Computer Vision–ECCV 2010*, pages 300–313, 2010.

[12] Forrester Cole, Phillip Isola, William Freeman, Frédo Durand, and Edward Adelson. Shapecollage: Occlusion-aware, example-based shape interpretation. *Computer Vision–ECCV 2012*, pages 665–678, 2012.

[13] Yingze Wang, Yu Chen, Jianzhuang Liu, and Xiaoou Tang. 3d reconstruction of curved objects from single 2d line drawings. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1834–1841. IEEE, 2009.

[14] Yu Chen, Jianzhuang Liu, and Xiaoou Tang. A divide-and-conquer approach to 3d object reconstruction from line drawings. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[15] Huizhong Chen, Sam S Tsai, Georg Schroth, David M Chen, Radek Grzeszczuk, and Bernd Girod. Robust text detection in natural images with edge-enhanced maximally stable extremal regions. In *Image Processing (ICIP), 2011 18th IEEE International Conference on*, pages 2609–2612. IEEE, 2011.

[16] Michael Kolomenkin, George Leifman, Ilan Shimshoni, and Ayellet Tal. Reconstruction of relief objects from line drawings. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 993–1000. IEEE, 2011.

[17] Christian Richardt, Jorge Lopez-Moreno, Adrien Bousseau, Maneesh Agrawala, and George Drettakis. Vectorising bitmaps into semi-transparent gradient layers. In *Computer Graphics Forum*, volume 33, pages 11–19. Wiley Online Library, 2014.

[18] James McCann and Nancy Pollard. Local layering. *ACM Transactions on Graphics (TOG)*, 28(3):84, 2009.

[19] James McCann and Nancy S Pollard. Soft stacking. In *Computer Graphics Forum*, volume 31, pages 469–478. Wiley Online Library, 2012.

[20] Jianchao Tan, Jyh-Ming Lien, and Yotam Gingold. Decomposing images into layers via rgb-space geometry. *ACM Transactions on Graphics (TOG)*, 36(1):7, 2016.

[21] James Herold and Thomas F Stahovich. A machine learning approach to automatic stroke segmentation. *Computers & Graphics*, 38:357–364, 2014.

[22] Jiri Matas, Ondrej Chum, Martin Urban, and Tomás Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and vision computing*, 22(10):761–767, 2004.

[23] Michael Donoser and Horst Bischof. Efficient maximally stable extremal region (mser) tracking. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 553–560. IEEE, 2006.

[24] Lluis Gomez and Dimosthenis Karatzas. A fast hierarchical method for multi-script and arbitrary oriented scene text extraction. *International Journal on Document Analysis and Recognition (IJDAR)*, 19(4):335–349, 2016.

[25] Emmanuel Prados and Olivier D Faugeras. " perspective shape from shading" and viscosity solutions. In *ICCV*, volume 3, page 826, 2003.

[26] Pierre-Louis Lions, Elisabeth Rouy, and A Tourin. Shape-from-shading, viscosity solutions and edges. *Numerische Mathematik*, 64(1):323–353, 1993.

[27] Emmanuel Prados and Olivier Faugeras. Unifying approaches and removing unrealistic assumptions in shape from shading: Mathematics can help. *Computer Vision-ECCV 2004*, pages 141–154, 2004.

[28] Ruo Zhang, Ping-Sing Tsai, James Edwin Cryer, and Mubarak Shah. Shape-from-shading: a survey. *IEEE transactions on pattern analysis and machine intelligence*, 21(8):690–706, 1999.

[29] Neil G Alldrin, Satya P Mallick, and David J Kriegman. Resolving the generalized bas-relief ambiguity by entropy minimization. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–7. IEEE, 2007.

[30] Micah K Johnson and Edward H Adelson. Shape estimation in natural illumination. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 2553–2560. IEEE, 2011.

[31] Yudeog Han, Joon-Young Lee, and In So Kweon. High quality shape from a single rgb-d image under uncalibrated natural illumination. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1617–1624, 2013.

[32] Daniel Sỳkora, Ladislav Kavan, Martin Čadík, Ondřej Jamriška, Alec Jacobson, Brian Whited, Maryann Simmons, and Olga Sorkine-Hornung. Ink-and-ray: Bas-relief meshes for adding global illumination effects to hand-drawn characters. *ACM Transactions on Graphics (TOG)*, 33(2):16, 2014.

[33] Katsushi Ikeuchi and Berthold KP Horn. Numerical shape from shading and occluding boundaries. *Artificial intelligence*, 17(1-3):141–184, 1981.

[34] Michael J Brooks and Berthold KP Horn. Shape and source from shading. 1985.

[35] Robert T. Frankot and Rama Chellappa. A method for enforcing integrability in shape from shading algorithms. *IEEE Transactions on pattern analysis and machine intelligence*, 10(4):439–451, 1988.

[36] Richard Szeliski. Fast shape from shading. *CVGIP: Image Understanding*, 53(2):129–153, 1991.

[37] Omar E Vega and Yee-Hong Yang. Shading logic: A heuristic approach to recover shape from shading. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(6):592–597, 1993.

[38] Qinfen Zheng and Rama Chellappa. Estimation of illuminant direction, albedo, and shape from shading. In *Computer Vision and Pattern Recognition, 1991. Proceedings CVPR'91., IEEE Computer Society Conference on*, pages 540–545. IEEE, 1991.

[39] Kyoung Mu Lee and C-CJ Kuo. Shape from shading with a linear triangular element surface model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(8):815–822, 1993.

[40] Jitendra Malik and Dror Maydan. Recovering three-dimensional shape from a single image of curved objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(6):555–566, 1989.

[41] Berthold KP Horn. Height and gradient from shading. *International journal of computer vision*, 5(1):37–75, 1990.

[42] Elisabeth Rouy and Agnès Tourin. A viscosity solutions approach to shape-from-shading. *SIAM Journal on Numerical Analysis*, 29(3):867–884, 1992.

[43] John Oliensis. Shape from shading as a partially well-constrained problem. *CVGIP: Image Understanding*, 54(2):163–183, 1991.

[44] John Oliensis and Paul Dupuis. A global algorithm for shape from shading. In *Computer Vision, 1993. Proceedings., Fourth International Conference on*, pages 692–701. IEEE, 1993.

[45] Paul Dupuis and John Oliensis. Direct method for reconstructing shape from shading. In *Computer Vision and Pattern Recognition, 1992. Proceedings CVPR'92., 1992 IEEE Computer Society Conference on*, pages 453–458. IEEE, 1992.

[46] Ron Kimmel and Alfred M Bruckstein. *Shape from shading via level sets*. Technion-Israel Institute of Technology. Center for Intelligent Systems; R 9209, 1992.

[47] Alex P Pentland. Local shading analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (2):170–187, 1984.

[48] Chia-Hoang Lee and Azriel Rosenfeld. Improved methods of estimating shape from shading using the light source coordinate system. *artificial Intelligence*, 26(2):125–143, 1985.

[49] Alex Pentland. Shape information from shading: a theory about human perception. *Spatial vision*, 4(2):165–182, 1989.

[50] Tsai Ping-Sing and Mubarak Shah. Shape from shading using linear approximation. *Image and Vision computing*, 12(8):487–498, 1994.

[51] Shree K Nayar, Katsushi Ikeuchi, and Takeo Kanade. Surface reflection: physical and geometrical perspectives. Technical report, CARNEGIE-MELLON UNIV PITTSBURGH PA ROBOTICS INST, 1989.

[52] Shree K Nayar, Katsushi Ikeuchi, and Takeo Kanade. Shape from interreflections. In *Computer Vision, 1990. Proceedings, Third International Conference on*, pages 2–11. IEEE, 1990.

[53] David Forsyth and Andrew Zisserman. Mutual illumination. In *Computer Vision and Pattern Recognition, 1989. Proceedings CVPR'89., IEEE Computer Society Conference on*, pages 466–473. IEEE, 1989.

[54] Henry Kang, Seungyong Lee, and Charles K Chui. Coherent line drawing. In *Proceedings of the 5th international symposium on Non-photorealistic animation and rendering*, pages 43–50. ACM, 2007.

[55] Thomas Porter and Tom Duff. Compositing digital images. In *ACM Siggraph Computer Graphics*, volume 18, pages 253–259. ACM, 1984.

[56] David Nistér and Henrik Stewénius. Linear time maximally stable extremal regions. *Computer Vision–ECCV 2008*, pages 183–196, 2008.

[57] Michel Couprie, Laurent Najman, and Gilles Bertrand. Algorithms for the topological watershed. In *Discrete geometry for computer imagery*, pages 172–182. Springer, 2005.

[58] Nicu D Cornea, Deborah Silver, and Patrick Min. Curve-skeleton properties, applications, and algorithms. *IEEE Transactions on visualization and computer graphics*, 13(3):0530–548, 2007.

[59] Yanlin Weng, Weiwei Xu, Yanchen Wu, Kun Zhou, and Baining Guo. 2d shape deformation using nonlinear least squares optimization. *The visual computer*, 22(9-11):653–660, 2006.

[60] Yeojin Kim, Byungmoon Kim, Jiyang Kim, and Young J Kim. Canvox: High-resolution vr painting in large volumetric canvas. *arXiv preprint arXiv:1704.02724*, 2017.

[61] Johannes Schmid, Martin Sebastian Senn, Markus Gross, and Robert W Sumner. Overcoat: an implicit canvas for 3d painting. *ACM Transactions on Graphics (TOG)*, 30(4):28, 2011.

[62] Pat Hanrahan and Paul Haeberli. Direct wysiwyg painting and texturing on 3d shapes: An error occurred during the printing of this article that reversed the print order of pages 118 and 119. while we have corrected the sort order of the 2 pages in the dl, the pdf did not allow us to repaginate the 2 pages. *ACM SIGGRAPH computer graphics*, 24(4):215–223, 1990.

[63] Paul Haeberli. Paint by numbers: Abstract image representations. In *ACM SIGGRAPH computer graphics*, volume 24, pages 207–214. ACM, 1990.

[64] Aaron Hertzmann. A survey of stroke-based rendering. Institute of Electrical and Electronics Engineers, 2003.

[65] Jingwan Lu, Pedro V Sander, and Adam Finkelstein. Interactive painterly stylization of images, videos and 3d animations. In *Proceedings of the 2010 ACM SIGGRAPH symposium on Interactive 3D Graphics and Games*, pages 127–134. ACM, 2010.

[66] Julie Daily and Kenneth Kiss. 3d painting: Paradigms for painting in a new dimension. In *Conference companion on Human factors in computing systems*, pages 296–297. ACM, 1995.

[67] George Katanics and Tasso Lappas. Deep canvas: integrating 3d painting and painterly rendering. *Theory and Practice of Non-Photorealistic Graphics: Algorithms, Methods, and Production Systems*, 10, 2003.

[68] Robert D Kalnins, Lee Markosian, Barbara J Meier, Michael A Kowalski, Joseph C Lee, Philip L Davidson, Matthew Webb, John F Hughes, and Adam Finkelstein. Wysiwyg npr: Drawing strokes directly on 3d models. *ACM Transactions on Graphics (TOG)*, 21(3):755–762, 2002.

[69] Sylvain Lefebvre, Samuel Hornus, Fabrice Neyret, et al. Octree textures on the gpu. *GPU gems*, 2:595–613, 2005.

[70] Daniel F Keefe, Daniel Acevedo Feliz, Tomer Moscovich, David H Laidlaw, and Joseph J LaViola Jr. Cavepainting: a fully immersive 3d artistic medium and interactive experience. In *Proceedings of the 2001 symposium on Interactive 3D graphics*, pages 85–93. ACM, 2001.