BOURNEMOUTH UNIVERSITY

TRANSFER DOCUMENT

# From Brush Painting to Bas-relief

*Author:*
Yunfei Fu

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

November 11, 2017

# ABSTRACT

Bas-relief is an art form part way between 3D sculpture and 2D painting. We present a new approach for generating a bas-relief from a single Chinese brush painting.We do not aim to recover exact depth of a painting, which is a tricky computer vision problem, requiring assumptions that are rarely satisfied.Instead, our approach exploits the concept of brush strokes, making each brush stroke possible to generate a correspond bas-relief proxies(depth map of brush strokes),and combine the depth map of brush strokes together to construct the bas-relief model. To segment brush strokes in 2D paintings, we apply layer decomposition and stroke segmentation by imposing boundary constraints, which can make strokes smooth and complete. The resulting brush strokes are sufficient to evoke the impression of the consistent 3D shapes on bas-relief, so that they may be further edited in 3D space. Currently, our research focus on Chinese brush paintings.We demonstrate our approach on a variety of Chinese brush paintings, and even some other suitable styles include rosemaling paintings and certain watercolor paintings, our approach is able to produce convincing bas-reliefs.

As a secondary application, we show how our brush stroke extraction algorithm could be used for image editing.And our brush stroke extraction algorithm is specifically geared to handling paintings with highly characteristic brushstrokes, in addition to Chinese paintings, other suitable styles include rosemailing paintings and certain van Gogh's oil paintings.

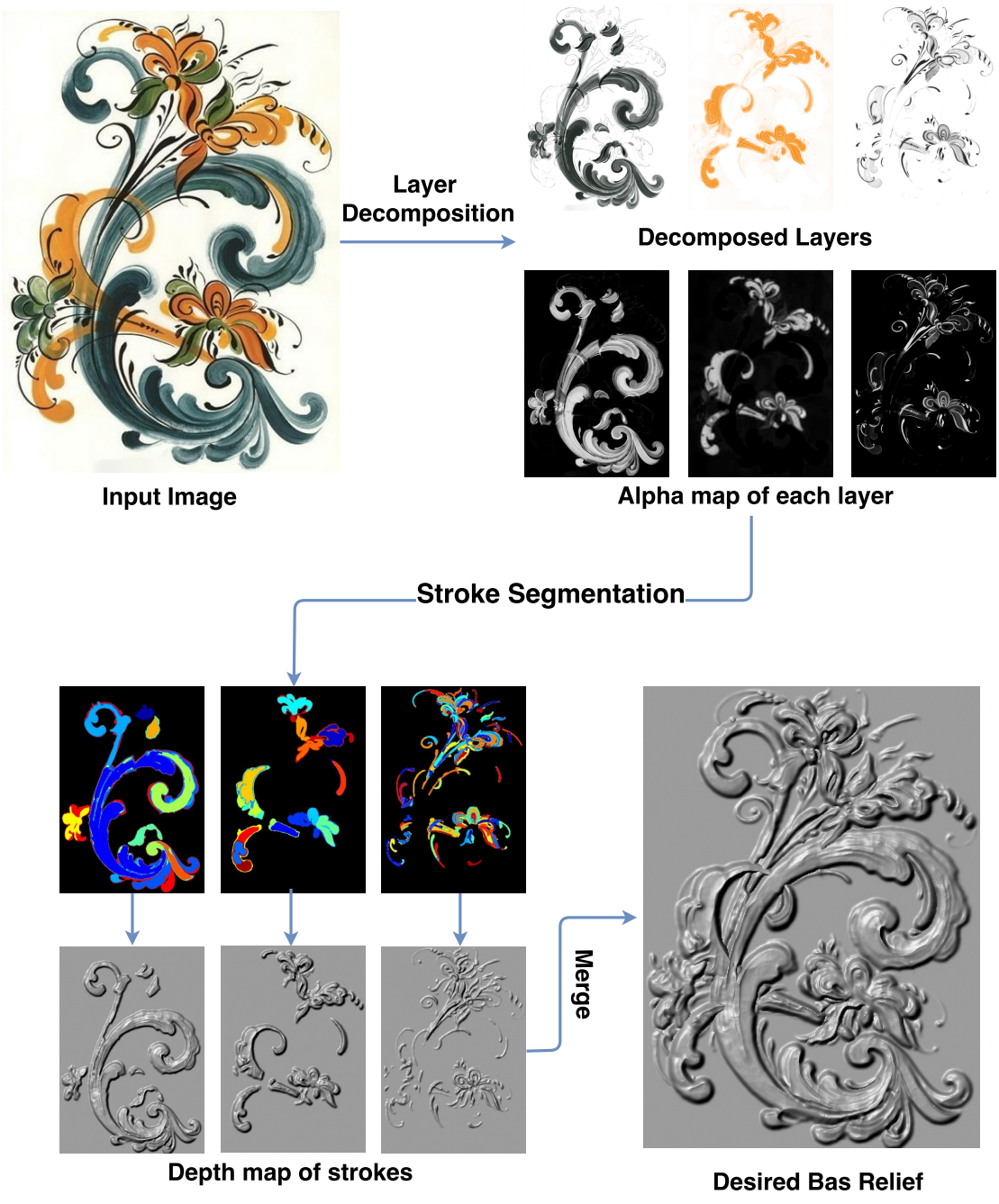



Figure 1: Pipeline Overview

# LIST OF BASIC TERMINOLOGY

**There are other terms to consider, but many of the important ones are explained in the report where relevant. These are terms not explicitly explained elsewhere.**

**Height/Depth map**: A depth map is an image or image channel that contains information relating to the distance of the surfaces of scene objects from a viewpoint.

**Intensity**: The measure of brightness of images. In most applications, measured between 0 (dark) and 255 (bright).

**Stroke**: A mark made by drawing a pen, pencil, or paintbrush across paper or canvas.

**Brush stroke**: A mark made by paintbrush across paper or canvas.

**Alpha map**: The alpha channel of an RGBA image ,which represents the opacity of each pixel.

# Contents

# List of Figures

# Chapter 1

# INTRODUCTION

## 1.1 Background

Relief is a kind of sculpture in which 3D sculpture are carved into a relatively flat surface. In essence, it creates a bridge between a full 3D sculpture and a 2D painting. On this spectrum,high relief is closest to full 3D, whereas flatter artworks are described as bas-relief. Among all the sculpture forms, bas-relief is the closest to 2D paintings[1] [2].

Bas-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. Today bas-reliefs are commonly found in a variety of media, commemorative medals, coins, souvenirs, 2.5D animation,and artistic sculptures for blind people. However, crafting bas-reliefs is a laborious, challenging and time consuming process. And with the commonly and cheaply available 3D printing facilities, there is a growing trend in the need of bas-relief art products.There are many ways to reach the same goals more easily with the help of computers. In the past two decades, a sizable amount of research has gone into developing bas-relief generation methods[3]. It is noteworthy that a 3D sculpture model can alway be represented by a 3D model, and a bas-relief are generally considered a depth map in previous studies.In this research, we use depth map to represent bas-relief surface as well.

Bas-relief is regarded as an art form part way between 3D sculpture and 2D painting [3][2][4][1][5].In general, the existing bas-relief generation methods can also be classified in two different categories with respect to their input[3]:

- 3D model based : using a 3D model (sculpture) as input
- 2D image based : using a 2D image as input

**3D model based:**
How to generate a bas-relief from a 3D sculpture? The 3D model based bas-relief generation methods focus on such a problem,in which a 3D sculpture are considered as a 3D digital model. The generation of bas-relief from 3D model was first studied in the pioneering work of [6], then various existing 3D model based methods have demonstrate how to compress 3D model into bas-relief [4][1][7][8] . However,this approach requires a 3D model as a starting point.

**2D image based:**
On the other hand, how to generate a bas-relief from a 2D painting?
There have been some bas-relief generation approaches available based on 2D image [9][10] and [11]. These approaches almost follow the "bas-relief ambiguity"[12], that is, roughly speaking, from a frontal view the sculpture looks like a full 3D object while a side-view reveals the disproportional depth. A 2D painting can be considered as a image,however,these image based methods are focusing on general photograph from real scene,assuming illumination and reflectance are known, and the image is formed from lighting and shading, which obviously unsuited for 2D paintings.And the generated bas-relief is a single depth map or mesh, which makes it hard for user to edit. Some research focus on bas-relief generation from 2D image of line drawing [13][14][15][16].However,line drawing based methods do not consider how to generate bas-reliefs with surface details: their approaches are limited to using information contained in a line drawing, which are not suited for paintings which contain information such as color,texture and stroke shape,etc. In general,the generated bas-relief are represented by a depth map or mesh.

As mentioned above, bas-relief is a art form part way between 2D painting and a full 3D sculpture[3][4][1][5][9] .And among all the sculpture forms, bas-relief is the closest to 2D paintings,as claimed by [1] [2], while how to generate bas-relief from artistic paintings remains a problem .

The aim of our research is to provide a method to generate a bas-relief from a Chinese painting. We also argue that because most Chinese paintings are produced with individual brush strokes, generating bas-relief surface from each brush stroke would preserve the original features of the painting.

A Chinese painting can be regarded as the union of brush strokes [17]. Most Chinese paintings are typically sparse with each brush stroke drawn very purposefully[18],and each stroke exists on its own as a piece of art[19]. Differing from the other bas-relief generation methods, our method will honor this very feature by generation bas-relief surfaces from the brush strokes individually. 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 brush stroke has to be faithfully extracted.
Second,unlike previous 2D image based methods focusing on photograph, a painting does not obey the rules of lighting and shading, which increase the difficulty to mimic the details on bas-relief surface.
Third, the generated bas-relief should be further editable allowing the artist to rearrange, tweaking and reshaping the bas-relief surface correspond to each brush strokes.

To extract the strokes from a brush painting, we need to identify and segment the overlapped strokes at first. We will then generate the depth map based on every strokes' opacity separately . All the brush strokes' depth map are finally merged together to yield the resulting bas-relief with the feature of original 2D painting preserved.

## 1.2 Motivation

The motivation of this research comes from three sides.
First,as mentioned above, bas-relief is regarded as a art form part way between 2D painting and a full 3D sculpture, as claimed by many previous works [3][4][1][5][9]. Research so far has been primarily done based on 3D model,and some other methods based on a single photograph or line drawing as input. However,how to generate bas-relief from artistic paintings remains a problem .
Second, due to the various styles of painting, it is difficult to come up with a general bas-relief generation method suitable for all 2D paintings. Some paintings may be too abstract for bas-relief generation.One the other hand, for traditional Western painting style developed in the Renaissance,which emphasizes realism[20], previous 2D image based methods may easily handle. Most Chinese paintings are typically sparse with each brush stroke drawn very purposefully[18], and such a feature can be found other painting styles,such as rosemailing painting and certain watercolor paintings. Study bas-relief generation from Chinese painting will naturally push forward the

research frontier of bas-relief generation from other painting.
Third, Bas-relief generation has wide applications in making objects such as commemorative medals, coins, souvenirs, and artistic sculptures for blind people.With the commonly and cheaply available 3D printing facilities, there is a growing trend in the need of bas-relief art products. On the other hand, the style and philosophies of Chinese paintings have influenced other painting styles [20], and as input of our bas-relief generation method, the digital image of Chinese paintings are easily accessible on the Internet.
Therefore, designing an efficient method for bas-relief generation from Chinese painting is important both theoretically and practically.

## 1.3 Contribution of Current Work

In this research, we propose a bas-relief generation method from a Chinese painting. As mentioned above, previous bas-relief generation methods have demonstrate how to generated bas-relief from 3D models,photographs and line drawings. In contrast, this research is the first attempt to generate bas-relief from a artistic painting.
Accompanying with this method, we have propose an algorithm for brush stroke extraction. To the best of our knowledge, there is lack of study on extracting brush strokes from paintings.The two most notable brush stroke extraction algorithms are proposed by [21] and [17]. [21] proposed a brush stroke extraction method by edge detection and clustering-based segmentation, but their method does not support overlap strokes.[17] proposed a algorithm for brush stroke extraction using segmentation techniques, however, this method requires professional artists to build a brush stroke library which makes it a challenging and time consuming process.
The major difference between our brush stroke extraction with the previous works is that our algorithm is based on a reformulated layer decomposition algorithm, which makes it capable of extraction overlapped brush strokes without the prior knowledge of brush strokes.
In contrast with previous 2D image based methods, our algorithm is the first attempt to generate bas-relief from opacity of brush strokes. And by generating bas-relief surface for each brush stroke , the combined final bas-relief is further editable allowing the artist to rearrange, tweaking and reshaping.

Our contributions include,
(1) Extraction of brush strokes. We develop a novel method to extract brush strokes based on palette color analysis and layer decomposition. Comparing with previous method brush stroke extraction methods, our method is capable of extracting overlapped brush strokes automatically without the prior

knowledge of brush strokes.
(2) Bas-relief generation of each brush stroke. We develop a novel method which may entirely generated every stroke to a correspond bas-relief surface(a depth map). By doing so, every single bas-relief surface of corresponding brush stroke can be editable, making it a useful tool for sculpture artists.
(3) Bas-relief generation from opacity. In contrast with previous 2D image based methods, our method use opacity instead of intensity to generate bas-relief, which can better preserve the feature of input painting.

Our experiments show that our method can effectively generate digital bas-reliefs for a range of input images, including Chinese paintings and some rosemailing paintings. We also demonstrate the utility of the decomposed brush strokes for image recoloring and image object insertion and animation. See section !!!

## 1.4 Document 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

## 2.1 Bas-relief Generation from 3D model

A significant amount of methods has been considered bas-relief generation from 3D models,[6] firstly proposed a method which creates bas-relief models by linearly compressing (squeezing) the depth map of 3D models. This method can not handle the depth gap between in 3D models and the output bas-reliefs can not successfully preserve the details on 3D models. Some researchers considered bas-relief generation as a geometry counterpart of the high dynamic range (HDR) image compression problem widely studied in computer graphics[7]. This approach calculates the differential coordinates of the input 3D models, then HDR image process method is applied to compress the 3D models. The output bas-relief can preserve salient feature and de-emphasize the others,but sometimes the bas-relief can be distorted and exaggerated. The methods proposed in [1][5] work on the gradient domain of depth map of 3D models, and rely on the combination of a saliency measure and a feature enhancement technique. these methods produce generally satisfactory bas-relief output,and preserve the features of 3D models, although again some areas can be over-emphasized. [8] generate the bas-relief with a optimized contrast-limited adaptive histogram equalization (CLAHE) method,in which the depth map of 3D model is compressed . Local contrast of depth map can be enhanced in this method,however, the detail preservation requires high resolution depth map from 3D model.

These 3D model based methods can often generate bas-relief with acceptable quality , but they all require inputs in the form of 3D models, which are often difficult and time-consuming to prepare, and obviously unsuited for bas-relief generation from 2D paintings.

## 2.2 Bas-relief Generation from 2D image

Automatic bas-relief generation from 2D image has recently become a significant research topic. Researches have explored recovering depth information from images specifically for the purpose of generating bas-reliefs.

### 2.2.1 Shape from shading(SFS) based methods

Some work uses shape from shading(SFS) algorithm for generating bas-relief from a photograph. SFS is a relative classic way for 3D shape recovery; see Zhang's survey [22]. 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[23], (1) orthographic SFS with a far light source [24]; (2) perspective SFS with a far light source[25]; (3) perspective SFS with a point light source at the optical center[23]; (4) a generic Hamiltonian. However, SFS is an ill-posed problem. The notable difficulty in SFS is the bas-relief ambiguity[12], 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[22] [26] [27] [2].

Unfortunately, SFS based methods is assuming illumination and reflectance are known, and the image is formed from lighting and shading, which works well for realistic photographs rather than paintings,simply apply SFS method is not enough to generate bas-relief with acceptable quality . see section !!!!!!

### 2.2.2 Line drawing based methods

Line drawing is a drawing made exclusively in solid lines. Rather than generating bas-relief from a photograph from real scene,some researches focus on bas-relief generation methods from 2D image of line drawing. Kolomenkin

et al.[13] aims to reconstruct a model from a complex line drawing that consists of many inter-related lines.At first, they extract the curves from line drawing.Then,junctions between lines are detected and margins are generated. By analyzing the connectivity between boundaries and curves, they reduce the problem to a constrained topological ordering of a graph. From these boundaries and curves with given depth, they use smooth interpolation across regions generate the bas-relief surface. Similarly,line labeling methods has been applied for shape construction from line drawings [14][15][16]. A labeling process would classify segmented lines into different labels, such as concave, convex and occluding, and these labels can give clues for the shape generation of bas-relief. [16] proposed a bas-relief generation method consists of six main steps: segmentation, completion, layering, inflation, stitching, and grafting.This method combines user indications and shape inflation to model smooth bas-relief shapes from line drawings.

However,line drawing based methods do not consider how to generate bas-reliefs with surface details: their approaches are limited to using information contained in a line drawing, which are not suited for brush strokes which contain information such as color,texture and stroke shape.

### 2.2.3 Gradient and intensity based methods

Wang et al.[28] demonstrate a method by constructing bas-reliefs from 2D images based on gradient operations. In their research image gradients were calculated, then by smoothing gradients to smooth shape changes. Finally , they boost fine features with user input masks. The height image was constructed modified height map. The pixel heights are compressed, and a triangle mesh representing the bas-relief is determined by placing a vertex at each pixel position. Most features can be preserve by proposed method, but no consideration is made for the front-to-back or overlapping relationship between different image regions.
[29] present a two-level approach for height map estimation from the rubbing images of restoring brick and stone relief. The relief is separated into low and high frequency components. The base relief of the low frequency component is estimated automatically with a partial differential equation (PDE)-based mesh deformation scheme. The high frequency detail is estimated directly from rubbing images automatically or optionally with minimal interactive processing. This method works well for reliefs based on stone rubbing images, but is unsuited to general photographs or paintings.

## 2.3 Brush 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.

Richardt et al.[30] present 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 . [31],[32]present two generalized layer decomposition methods, which allow pixels to have independent layer orders and layers to partially overlap each other.[33] present a layer decomposition method based on RGB-space geometry. They assume that all possible image colors are convex combinations of the palette colors. Computing the convex hull of image colors and per-pixel layer opacities is converted into a convex optimization problem. Thus, method in [33] can work well without prior knowledge of palette colors. However the proposed method simply focus on layer decomposition not stroke segmentation, how to extract the brush stroke remains a problem.

Li et al.[21] describe a method based on seed growing for brush stroke segmentation.Starting from seed coordinates, neighboring pixels are visited by exploiting a region-growing-based approach with a variable threshold, which is initialized at a predefined and automatic updated by a shape validation method. But their brush stroke segmentation is limited to finding brush strokes in Van Gogh's paintings, and it does not support segmenting overlapped strokes.

Xu et al.[17] aims at decomposing Chinese paintings into a collection of layered brush strokes, with an assumption that at most two strokes are overlapping. However, their approach requires a good amount of prior knowledge of shape and order of strokes.The segmentation of brush strokes is based on using a brush stroke library. While our method needs no brush stroke library, and our segmentation is based on multiple layers which more than two brush strokes could overlap each others(Figure 5.2).

Most Stable Extremal Regions (MSERs) algorithm[34] has been proved to be a very efficient way in stroke segmentation from scene text images[35][36], and the revised version has been widely used in stroke segmentation of handwritten characters[37].

The Most Stable Extremal Regions (MSERs) algorithm[34] was used for establishing correspondence in wide-baseline stereo.[38] 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, see Section 5.1.

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.
(3) Moreover, like the other existing segmentation approaches, the MSERs algorithm encounters over-segmentation issue as well.
To tackle these challenges, the coherent lines method [39] is introduced into MSERs in our algorithm, which both enhances the edges of strokes and preserves the completeness of strokes, see Section 5.2.

# 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 palette colors. In our decomposition, each layer represents a single-color coat of paint applied with varying opacity. Secondly, each resulting layer is further segmented into multiple regions which represent brush strokes respectively.As shown in Figure 3.1 ,we use different colors to show the results of our brush stroke extraction, and each color represents a single extracted brush stroke. 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 this research, we assume a brush stroke only utilizes a single palette color. Therefore, layer decomposition helps classify brush strokes separately into different layers based on their colors so that every layer contains the brush strokes which are better separated. However, wrong layer decomposition may cut one stroke into two or more layers. It is observed that most brush strokes in Chinese paintings follow regular patterns. For instance, the color and transparency change very little in the direction of the stroke [17]. So, we introduce the edge tangent flow (ETF) field and the coherent line [39] 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 maximally stable extremal regions(MSER) algorithm [38] again for extracting spurious edges within one

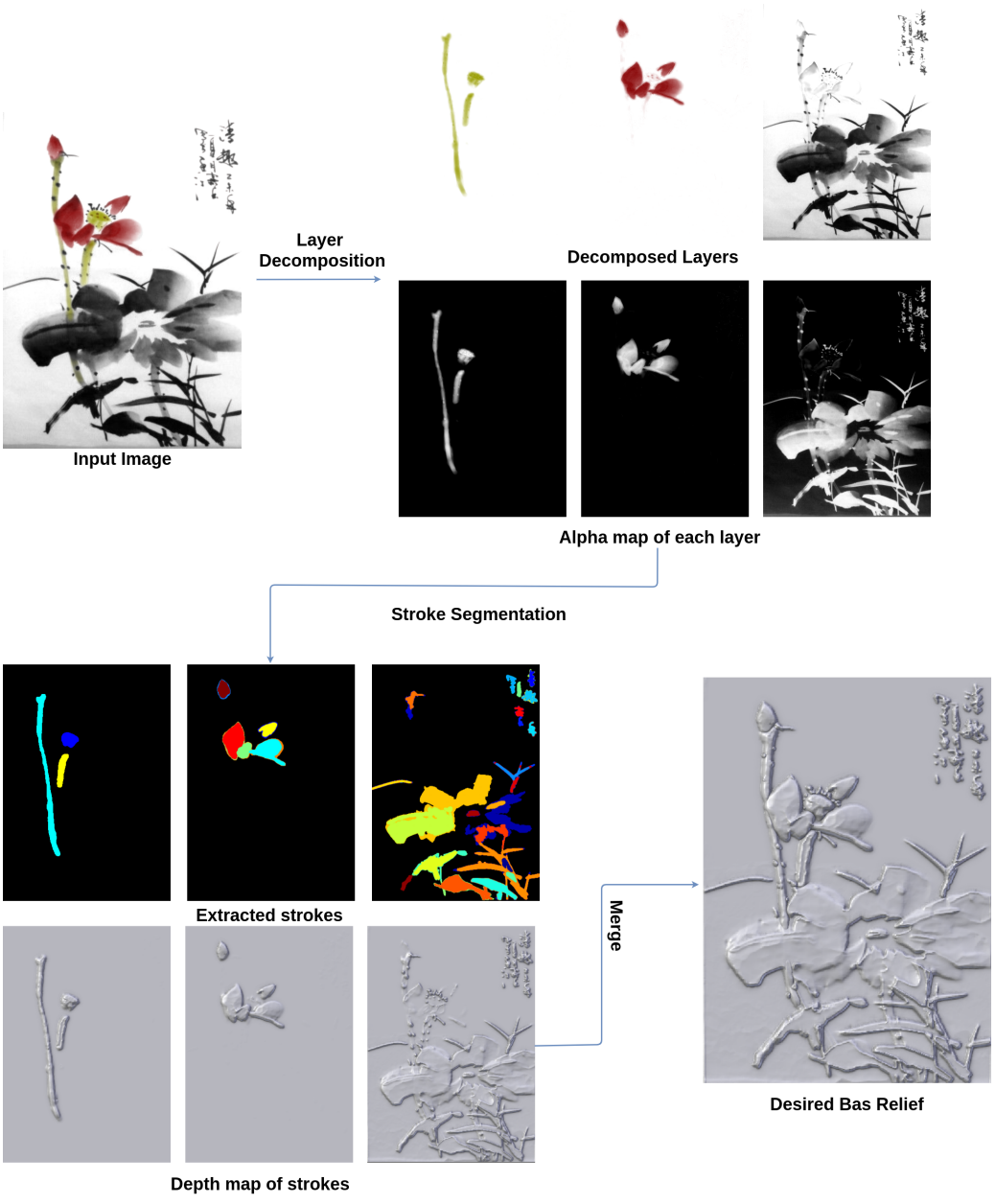



Figure 3.1: OVERVIEW OF PROPOSED METHOD

layer. Furthermore, to generate the depth maps of the strokes individually, we perform shape from shading(SFS) 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 4.8 for details). 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 can be further edited,since the user may want to reform the bas-relief models for secondary creation.

# Chapter 4

# LAYER DECOMPOSITION

Digital painting with different layers is an integral feature of digital image editing software. However, layers may never have existed for a scanned physical painting.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.So,before extracting brush strokes, we decompose a Chinese painting into layers.
In our decomposition, 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 [33] by involving the coherent lines [39] in our implementation. In the following we first address the layer decomposition algorithm [33] briefly and then discuss our modification.

## 4.1 Identify Paint Colors

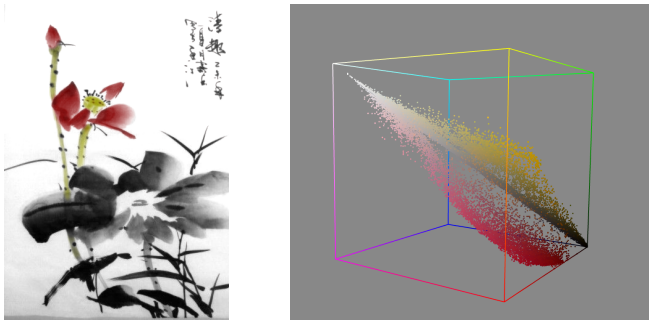


Figure 4.1: Geometry of input image pixels in RGB-space

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 hull of in the RGB space as showed in Figure 4.1 and Figure 4.2 , and every vertice is considered as a paint color. And base on such idea, we can represent each pixel color based on painted colors:

$$p = \Sigma \omega_i c_i$$

$p$ 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 [33]. 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 layers. In Tan's work [33] 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. And based on such method ,we generate a palette of paint colors of the input paintings.

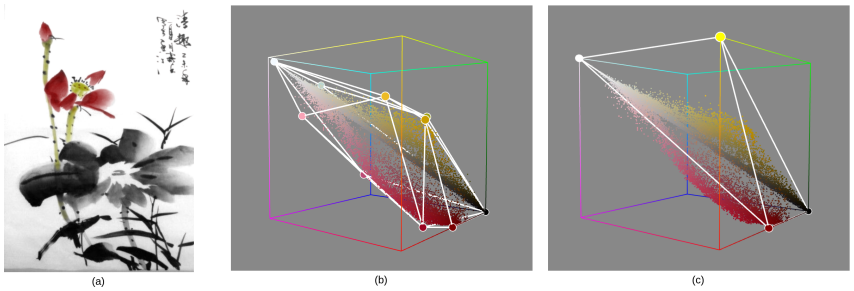



Figure 4.2: Convex hull of input image

(a) A simple digital painting's (b) convex hull in RGB-space is complex due to rounding. (c) The result of simplification algorithm

## 4.2 Layer Decomposition Scheme

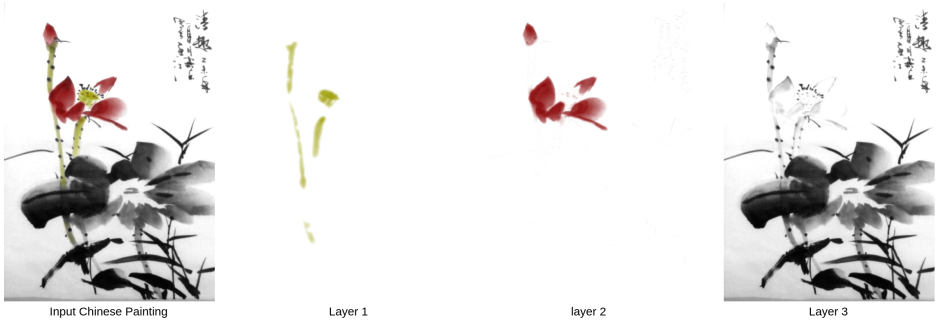



Figure 4.3: Decomposition layers

The standard Porter-Duff "A over B" compositing operation in [40] is described that when the pixel $A$ with color $A_{RGB}$ and translucency $\alpha_A$ is placed over the pixel $B$ with color $B_{RGB}$ and translucency $\alpha_B$, 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. 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 based on the "A over B" compositing [40],

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

where $C_i$ denotes the i-th layer's color, is the opacity of $\alpha_i$ , 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,

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

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,

$$E = \omega_{polynomial}E_{polynomial} + \omega_{opaque}E_{opaque} + \omega_{spatial}E_{spatial} \qquad (4.1)$$

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

### 4.2.1 Our Modified Layer Decomposition

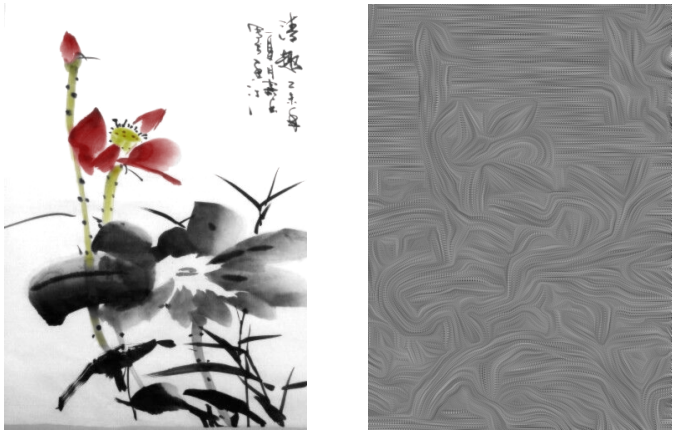


Figure 4.4: Edge Tangent Flow

As we can see in Figure 4.3,the extracted layers failed to maintain the completeness of the brush strokes, especially in the first layer. To enhance the smoothness and completeness of extracted strokes, the coherent line drawing technique in [39] is introduced to Eq 4.1 ,which is a flow-guided anisotropic filtering framework.In the coherent line drawing technique, the edge tangent flow(ETF) of a image is calculated, represented by a vector $t^{new(x)}$. Here, we use vector $t^{new(x)}$ to mimic the brush stroke direction. Figure 4.4 shows the edge tangent flow (ETF) field of a Chinese painting.
The ETF field is defined as,

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

$t(x)$ 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 $\omega_s$ employs the radially-symmetric box filter with some radius. The magnitude weight function $\omega_m$ 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, $\omega_d$, 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 4.5: Comparison of layer decomposition

Comparison of decomposed first layer before and after modification. Left shows the original result of Eq 4.1 , right shows the result by using $E_{flow}$ instead of $E_{spatial}$ 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 $\theta$ 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.

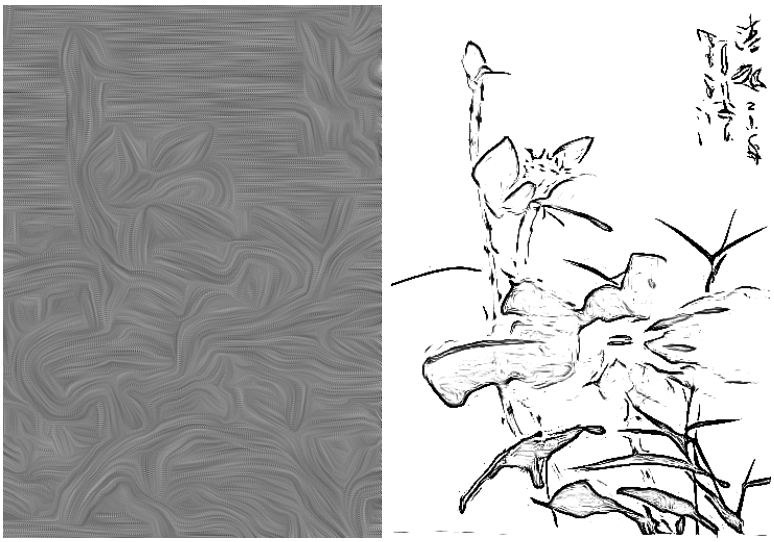


Figure 4.6: Edge Tangent Flow field and coherent lines of the painting.

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.

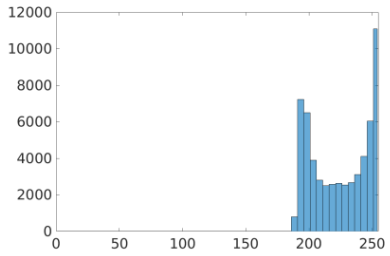
Figure 4.7: Comparison of layer decomposition before and after using $E_{egde}$ in Eq 4.1;

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.

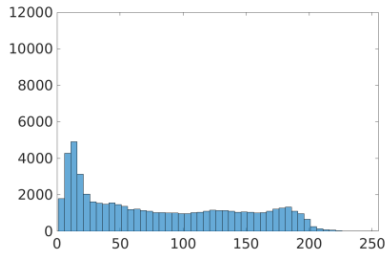
It is most likely to make the highest contrast in the perpendicular direction, that is, the gradient direction. When moving along the edge flow, the DoG filter is applied in the gradient direction. As a result, we can 'exaggerate' the filter output along genuine edges, while 'attenuating' the output 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) intensity map of layer2

(b) alpha map of layer2




(c) histogram of intensity map of layer2




(d) histogram of alpha map of layer2

Figure 4.8: Comparison between transparency and intensity of layer2

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, 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 $\alpha_i$ 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 $\omega_{flow} = 100, \omega_{edge} = 20$ for all our examples. Figure 8 shows an evaluation of the effect of changing weights.

To demonstrate our results can better preserve the smoothness and completeness of brush strokes, we perform the schemes of Eq 4.1 and Eq 4.5 separately on the same set of Chinese 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.

To demonstrate the robustness of our approach, including Chinese paintings, we also perform the comparison on some other paintings with highly characteristic brush strokes, such as rosemailing paintings[41] and Vincent van Gogh's paintings[21].

**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

To generated bas-relief model from brush strokes,we need to first extract brush strokes.In this chapter, we demonstrate how to extract brush strokes from generated layers by modified Maximally Stable Extremal Regions (MSERs) algorithm, and we show that our optimization is more suitable for brush stroke extraction. In this chapter, the default MSERs algorithm and our modified algorithm are explained and their main characteristics compared. By successfully extract brush strokes, our algorithm can also be used in image editing,see section 5.3.

The Maximally Stable Extremal Regions (MSERs) proposed in [38] and [42]for feature detection and image segmentation. A MSER is a 2D region detected by MSERs algorithm. Generally,a brush stroke is considered as a closed stable region on paintings and it might be painted on different scales. MSERs have properties that form their superior performance as stable segmentation. The segmented MSER is closed under continuous geometric transformations and is invariant to affine intensity changes. Furthermore MSER are detected at different scales. So, we employ the Maximally Stable Extremal Regions (MSERs) proposed in [38] and [42] to extract brush strokes. MSERs algorithm requires a distinct difference between background and foreground while allowing a small variation of intensity within the selected stroke region. Usually, the brush strokes on the decomposed layers satisfy this requirement. However, MSERs may fail in segmentation with the following scenarios,

(1) The brush stroke with the intensity very close to the background;

(2) Two adjacent brush stroke painted by different colors with the similar intensity;

(3) Overlapped brush strokes.

(4) Moreover, like the other existing segmentation approaches, the MSERs algorithm encounters over-segmentation issue as well.
To tackle these challenges, the coherent lines [39] 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

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,

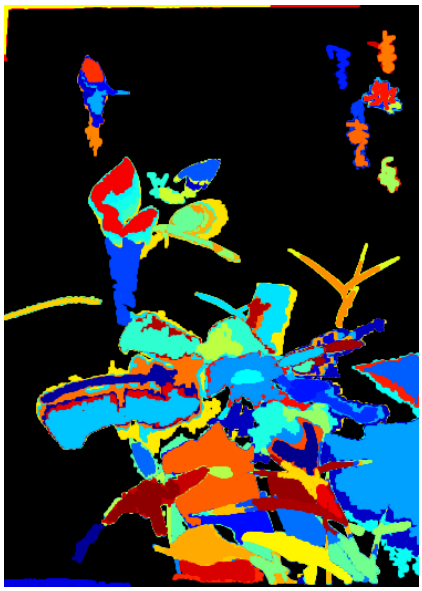
$$\forall p \in S, \forall q \in \partial S, \longrightarrow I(p) \geq I(q)$$

where $\partial S$ denotes the boundary of $S.p$ and $q$ represent different pixels on the image, and $I(x)$ represent the intensity value of pixel $x$. 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

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

where $|.|$ denotes the cardinality, $S_i^g$ is the i-th region which is obtained by thresholding at an intensity value g and $\Delta$ 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 $\gamma$, which is a local minimum along the path to the root of the tree.

(a) input image



(b) MSERs of the input image Perform the original MSERs on the intensity of image,different color indicates different regions.

## 5.2 Modified MSERs Algorithm

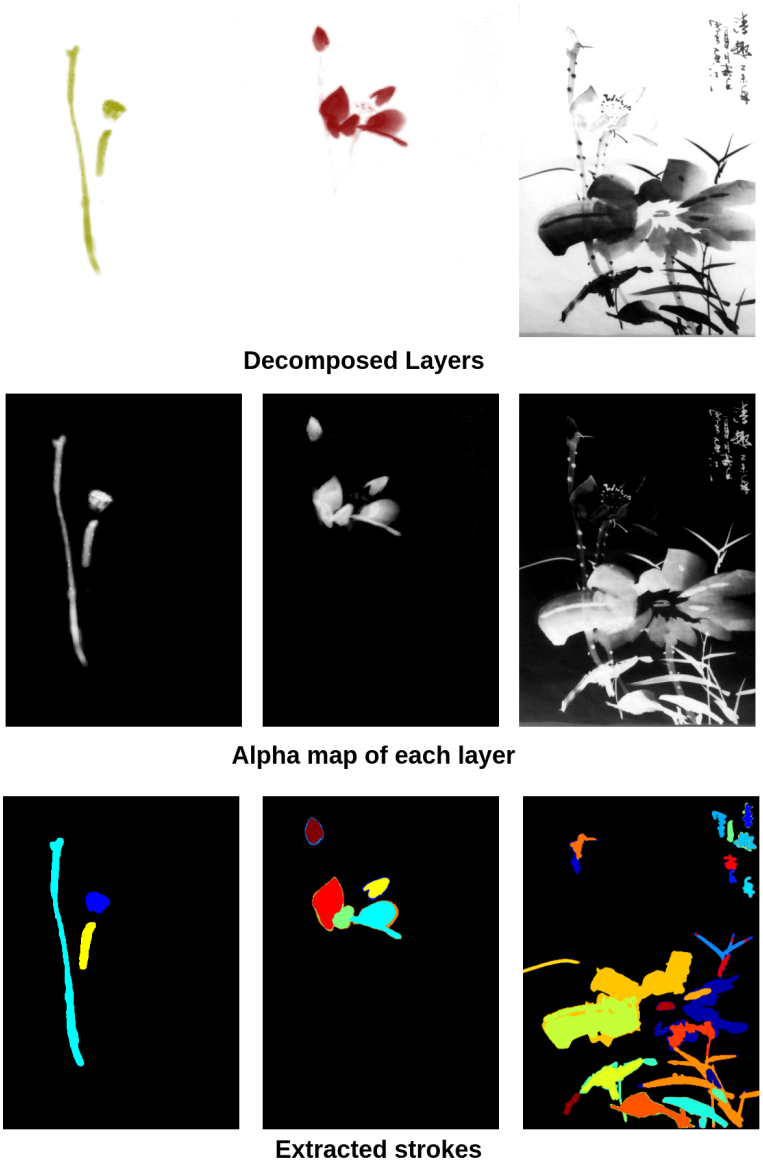



Figure 5.2: The opacity maps, and MSER regions of three layers by the modified MSERs.

In terms of the definition of the area change rate $\gamma$, MSERs may fail in segmentation with the following scenarios,
(1) The brush stroke with the intensity very close to the background;

(2) Adjacent brush strokes painted by different colors with the similar intensity;
(3) Overlapped brush strokes with different color.
(4) Moreover, like the other existing segmentation approaches, the MSERs algorithm encounters over-segmentation issue as well.
Comparing with default MSERs algorithm use intensity of input, our extraction is performed on opacity of generated layers.By doing so,in each layer, the opacity of the background is close to zero, which can be easily filter out, see Figure 5.2 , so, the first scenario can be handled.
The opacity of the brush strokes is always independent of the color,so two adjacent brush stroke painted by different colors would be separated into different layers, so,naturally,our extraction is suitable for the second scenario and third scenario, see Figure 5.2.
Moreover, 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 4.8. It can be noted that the opacity of the layer contains richer layered details than the intensity. So,our first modification is to perform MSERs on the opacity of every layer.
Secondly,we aim at the third scenarios of over-segmentation. 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 $\gamma$ as,

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

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 $\partial S$ is occupied by coherent lines.

Figure 5.2 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, and no region is selected from background.

## 5.3 Image editing based on brush strokes

Once the brushstrokes are extracted,we can simply recoloring strokes by assign a RGB value to the brushstroke while keep its opacity value. Figure 5.3 a and 5.3 b shows recoloring strokes on three paintings respectively. As the strokes have been extracted, it is easy to separately recolor one or more strokes with different colors. However, for van Gogh oil painting, it seems tricky, since the painting is composed of lots of brushstrokes. Thus, we only recolor all the brushstrokes extract in one layer. Moreover, Figure 10 further illustrates the brushstrokes of the van Gogh oil painting, which justifies the efficiency of our brushstroke extraction method.

Figure 5.3 c shows stroke manipulation through inserting objects . One of image editing tasks is to change a specified region with a new object in a seamless and effortless manner. Here we are interested in inserting new objects into a painting while keeping the transparency of the painting. Note that the inserted objects are opaque and are inserted between brushstrokes here. The occluded regions of the objects are visible due to the transparency of the brushstrokes. Unlike the traditional image synthesis approaches, our implementation works on the strokes, which both guarantees seamless and keep the transparency of brushstrokes on the painting.

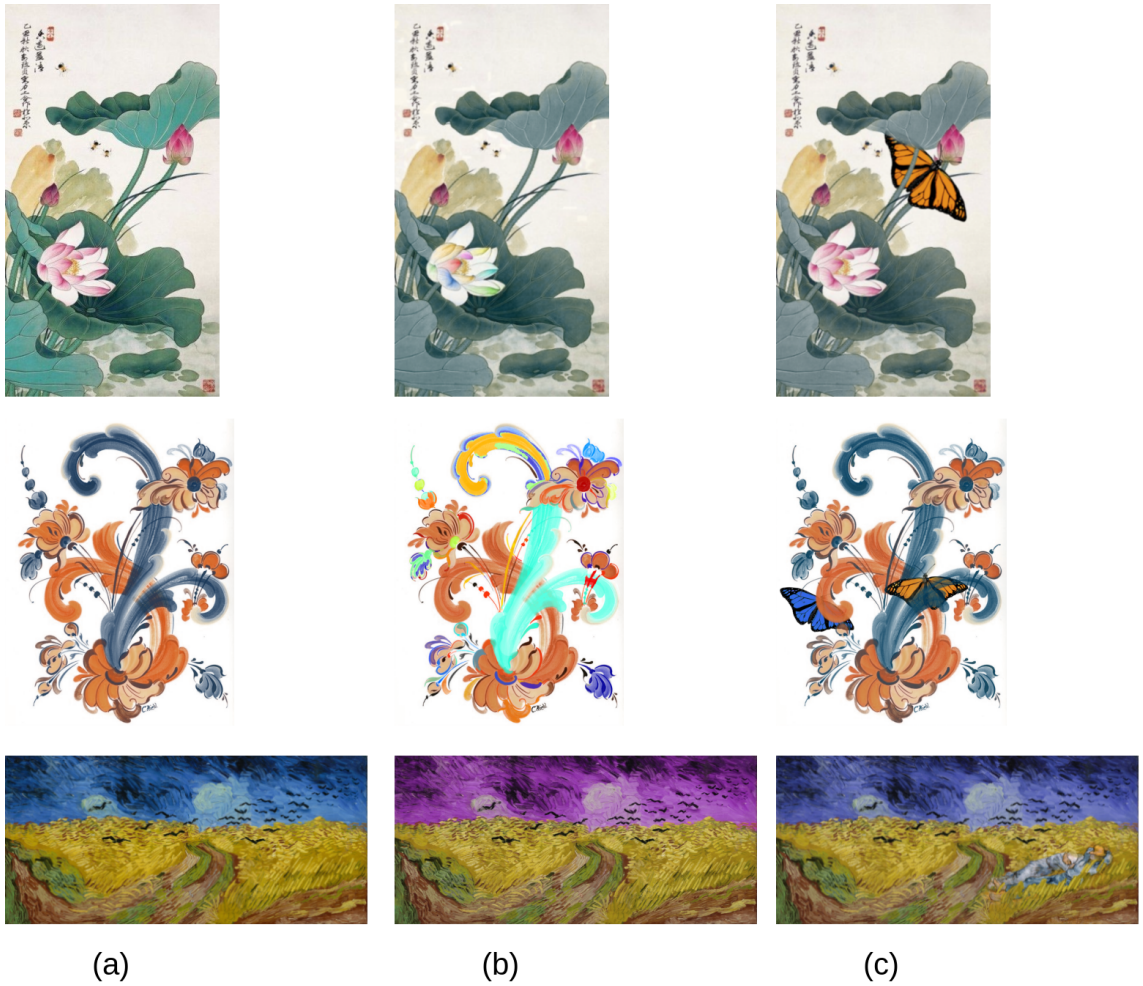



Figure 5.3: Image editing based on brush strokes

Column a) shows 3 kinds of brush paintings, Chinese painting ,Rosemailing,and van Gogh oil painting . Column b) shows recoloring multiple strokes. Herein, for van Gogh oil painting, we show recoloring all the brushstrokes in one layer. Column c) shows inserting objects into these 3 paintings, in which the objects are opaque and are inserted in between brushstrokes.

# Chapter 6

# BAS-RELIEF BASED ON BRUSH STROKE

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, see Figure 3.1.It is noteworthy that a depth map can represent a bas-relief model. In this Chapter, we demonstrate how to generate bas-relief surface from brush strokes, and how it can be modified. We also compare our algorithm against the two most notable 2D image based bas-relief generation algorithm.

In our implementation, we employ the orthogonal SFS [25] on the segmented strokes. These extracted strokes take into account the spatial occlusion and are suitable for 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 shape from shading algorithm 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 color intensity (see Figure 4.8). Each stroke has an appropriate distribution of opacity, which is

in favor of a layered look, so we reformulate the equation :

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

$\alpha(x)$ is the opacity value of pixel $x$ on a brush stroke. To make the bas-relief more inflated,we rewrite the as,

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

where $\Delta$ is a positive displacement. This modification may make the surface inflated. By solving this equation, we can generated a depth map for each brush stroke. As show in Figure 6.1, we input the opacity map of a decomposed layer, and from that layer we extract three brush strokes shown in different color,and by apply our algorithm we can generate bas-reliefs(depth maps) from each one of the brush strokes on this layers, and by generate bas-reliefs from all brush strokes, and merge them together, we automatically generate a bas-relief from the input Chinese painting. To differentiate the bas-relief generated form a single stroke and the final combined bas-relief, we use the term "stroke-relief" to indicate a bas-relief(depth map) generated from a single stroke. And just like a Chinese painting can be regarded as a union of brush strokes [17] ,in our approach, a bas-relief can be regarded as a union of stroke-relief.

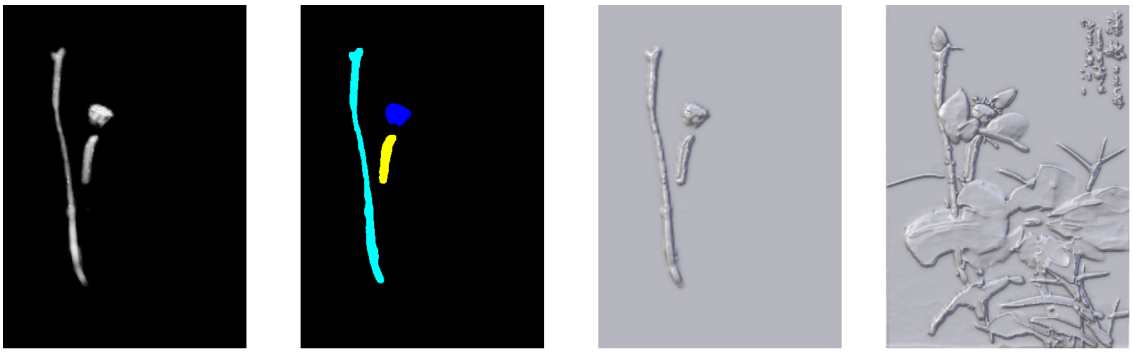



Figure 6.1: Bas-relief generation from brush strokes

## 6.1 Comparison

This section compares our algorithm against the two most notable 2D image based bas-relief methods,

The image-based bas-relief approaches[9] [13] perform shape from shading algorithm 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 orthogonal SFS method [25] on the segmented Rosemaling painting shown in Figure 4.4 to generate the depth maps. It can be seen that there are lots of over-segmentation, or even incorrect segmentation as shown in Figure **??**.

## 6.2 Manipulate Bas-relief of stroke

Moreover, it is desirable to allow the users to manipulate the brush stroke models for secondary creation purposes. We present three approaches based on the generated bar-relief 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 6.2. 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.

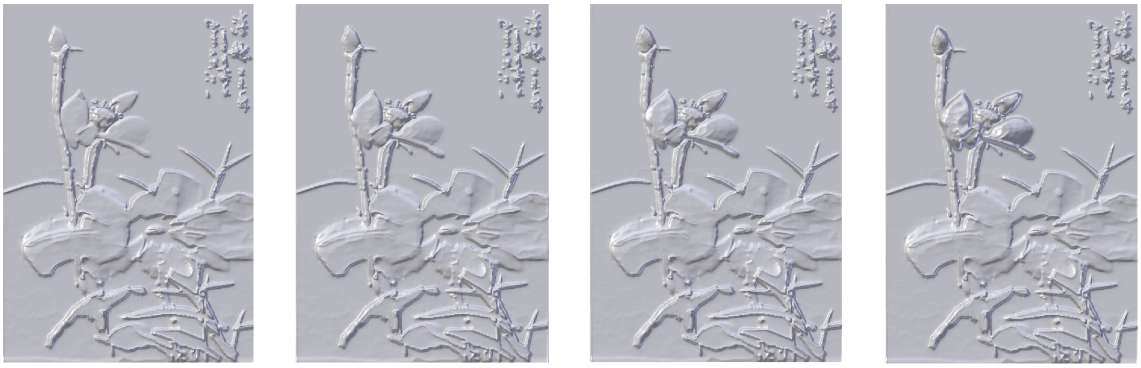


Figure 6.2: Rising strokes

The selected strokes 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 6.4. 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 6.4. We can further change the shape of these 3D strokes if needed.

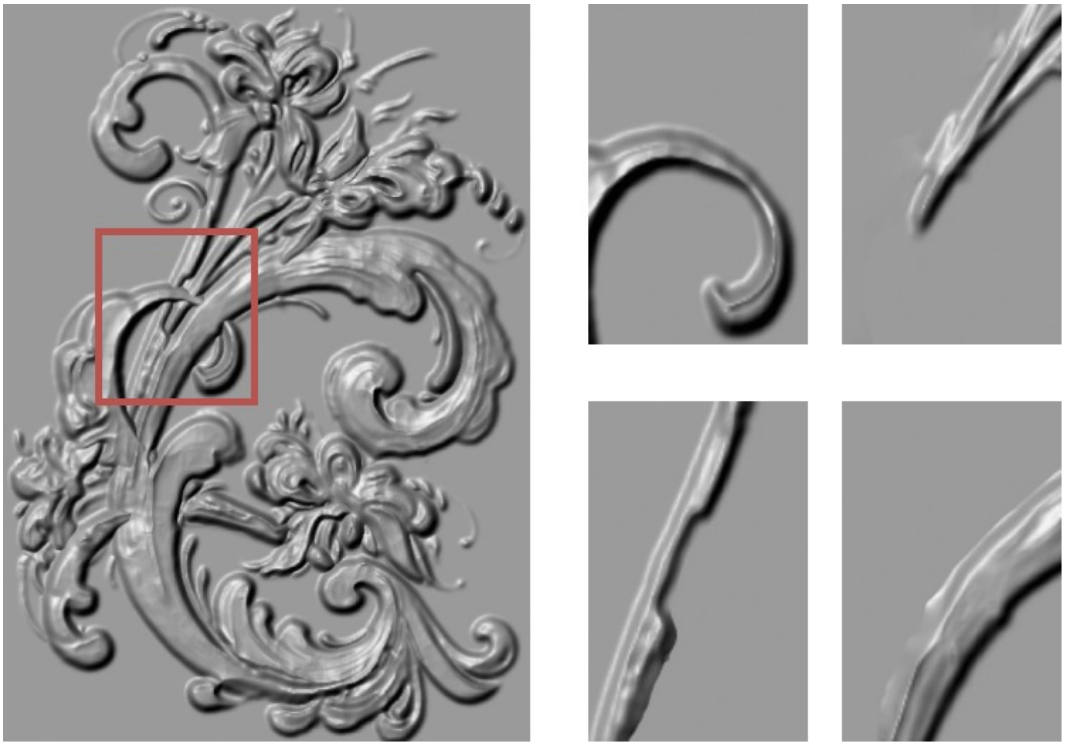


Figure 6.3: Strokes segmented in selected region

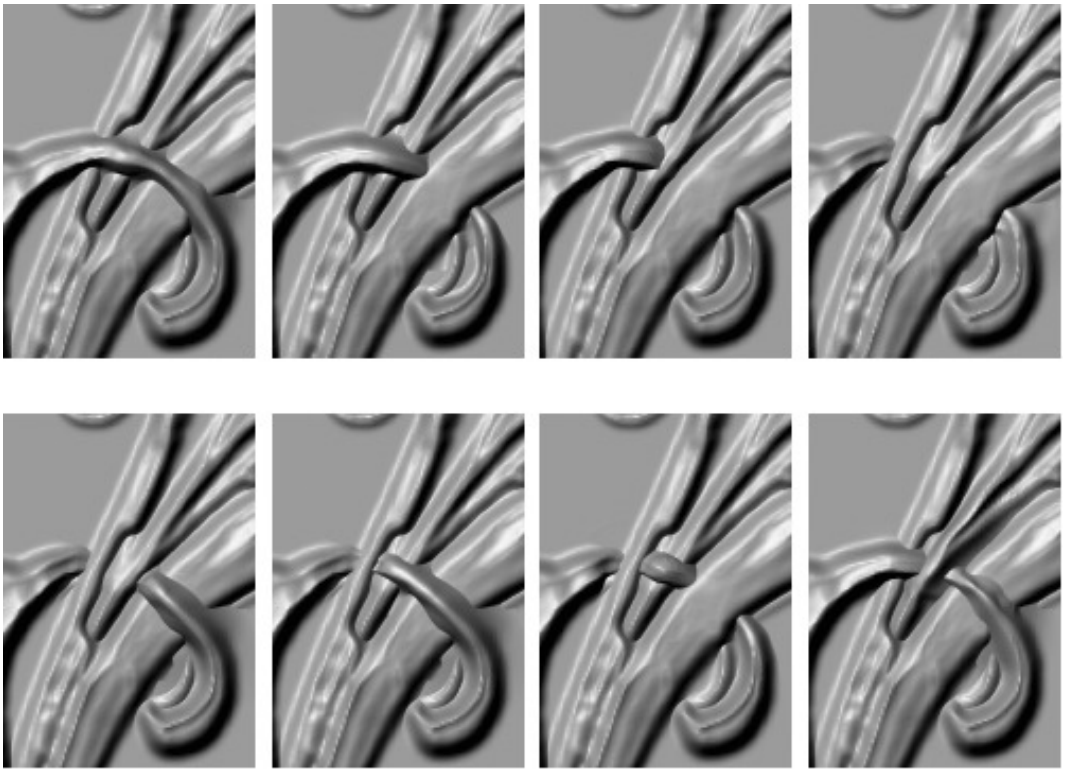


Figure 6.4: The result of stitching strokes

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 $\partial S_i$, 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 [43]. 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 [44] 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 $\partial S_i$; 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 6.5 shows how to change the shape of the model of $S_i$.

(a) selected stroke (b) modified depth map

Figure 6.5: 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 modified 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),the differences between our algorithm and the available codes have been explain in Section 4.2.1 and 5.2. We 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 [33] and [42], 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 |

# Bibliography


[1] 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.

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

[3] Zachary Benzaid. *Analysis of Bas-Relief Generation Techniques*. PhD thesis, The University of Wisconsin-Milwaukee, 2017.

[4] 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.

[5] Jens Kerber, Meili Wang, Jian Chang, Jian J Zhang, Alexander Belyaev, and H-P Seidel. Computer assisted relief generation—a survey. In *Computer Graphics Forum*, volume 31, pages 2363–2377. Wiley Online Library, 2012.

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

[7] Wenhao Song, Alexander Belyaev, and Hans-Peter Seidel. Automatic generation of bas-reliefs from 3d shapes. In *Shape Modeling and Applications, 2007. SMI'07. IEEE International Conference on*, pages 211–214. IEEE, 2007.

[8] Xianfang Sun, Paul L Rosin, Ralph R Martin, and Frank C Langbein. Bas-relief generation using adaptive histogram equalization. *IEEE transactions on visualization and computer graphics*, 15(4):642–653, 2009.

[9] 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.

[10] 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.

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

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

[13] 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.

[14] Peter AC Varley and Ralph R Martin. Estimating depth from line drawing. In *Proceedings of the seventh ACM symposium on Solid modeling and applications*, pages 180–191. ACM, 2002.

[15] Jitendra Malik. Interpreting line drawings of curved objects. *International Journal of Computer Vision*, 1(1):73–103, 1987.

[16] 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.

[17] 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.

[18] Ray Smith and EJ Lloyd. Art school, 1997.

[19] Ross B Girshick. *Simulating Chinese brush painting: the parametric hairy brush*. ACM, 2004.

[20] Nelson SH Chu and Chiew-Lan Tai. Real-time painting with an expressive virtual chinese brush. *IEEE Computer Graphics and applications*, 24(5):76–85, 2004.

[21] Jia Li, Lei Yao, Ella Hendriks, and James Z Wang. Rhythmic brushstrokes distinguish van gogh from his contemporaries: findings via automated brushstroke extraction. *IEEE transactions on pattern analysis and machine intelligence*, 34(6):1159–1176, 2012.

[22] 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.

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

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

[25] 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.

[26] 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.

[27] 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.

[28] Meili Wang, Jian Chang, Junjun Pan, and J Zhang. Image-based bas-relief generation with gradient operation. In *Proceedings of the 11th IASTED International Conference Computer Graphics and Imaging, Innsbruck*, volume 679, page 33, 2010.

[29] Zhuwen Li, Song Wang, Jinhui Yu, and Kwan-Liu Ma. Restoration of brick and stone relief from single rubbing images. *IEEE Transactions on Visualization and Computer Graphics*, 18(2):177–187, 2012.

[30] 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.

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

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

[33] 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.

[34] 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.

[35] Lukas Neumann and Jiri Matas. Text localization in real-world images using efficiently pruned exhaustive search. In *Document Analysis and Recognition (ICDAR), 2011 International Conference on*, pages 687–691. IEEE, 2011.

[36] Lluis Gomez and Dimosthenis Karatzas. Multi-script text extraction from natural scenes. In *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*, pages 467–471. IEEE, 2013.

[37] 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.

[38] 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.

[39] 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.

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

[41] Nils Ellingsgard, Unn Plahter, and Erling Skaug. Rosemaling i hallingdal. 1978.

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

[43] 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.

[44] 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.