BOURNEMOUTH UNIVERSITY

TRANSFER DOCUMENT

From Chinese Painting to Bas-relief

Author:
Yunfei Fu

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

November 29, 2017

ABSTRACT

Bas-relief is an art form part way between 3D sculpture and 2D painting automatically. We present a novel approach for generating a bas-relief from a single Chinese 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 combines the depth maps of brush strokes together to build the bas-relief model. To segment brush strokes in 2D paintings, we apply layer decomposition and stroke segmentation by imposing boundary constraints. The resulting brush strokes are sufficient to preserve the feature of input Chinese painting. Currently, our research focus on Chinese paintings.We demonstrate our approach is able to produce convincing bas-reliefs from a variety of Chinese paintings(with human, animal, flower,etc.), and even some other suitable styles include rosemaling paintings.

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 each brush stroke drawn very purposefully, in addition to Chinese paintings, other suitable styles include rosemailing paintings and Vincent van Gogh's oil paintings.

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 drawing paintbrush across paper or canvas.

Opacity/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 method of sculpture in which forms 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].

Bas-relief is regarded as an art form part way between 3D sculpture and 2D painting [3][2][4][1][5]. Correspondingly, 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

1.1.1 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 is 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 demonstrated how to compress 3D model into bas-relief [4][1][7][8] . However,this approach requires a 3D model as a starting point.

1.1.2 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 a 2D image [9][10] [11][12]. These approaches almost follow the "bas-relief ambiguity"[13], 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 input 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 a line drawing [14][15][16][17]. However,line drawing based methods generate bas-relief without consideration for surface details: their approaches are limited to using information contained in a line drawing, which is not suited for paintings containing information such as color,texture and stroke shape.

Reconstructing a surface given a single 2D image is an ill-posed problem in general. As described in most of the papers on 2D image based algorithms, the problem that manifests itself immediately is that there is no complete knowledge of the depth within a single image.However, with the purpose of generating bas-relief with acceptable quality, there are several factors to be considered specifically for the framing of the problem.[3] describes the some core important factors in the bas-relief generation process: depth information, silhouettes and edges,fine details.
Depth information: Within a single image there is no complete knowledge of the depth. A reasonable interpretation must be determined in some manner.
Silhouettes and Edges: Generally, silhouettes and edges are used for separate object and background and generating a reasonable contrast between

the foreground objects and the background as a starting establishment for interpreting heights.
Fine Details: Many images will contain many small features and details within the subject of the image. In the process of generate bas-relief from 2D images,small features and details are preferable to maintain.Examples are features like texture of a hand or the shininess of an apple.
In our work, we evaluate our result and compare with other alternative methods based on these three factors, see Section 6.1.

1.2 Aim and Challenges

As mentioned above, bas-relief is considered as 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 paintings remains a problem .
The aim of our research is to provide a method to generate a bas-relief from a Chinese painting automatically. 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 [19]. Most Chinese paintings are typically sparse with each brush stroke drawn very purposefully[20],and each stroke exists on its own as a piece of art[21]. Differing from the other bas-relief generation methods, our method will preserve this very feature by generating bas-relief surfaces from the brush strokes individually. This however demands to conquer several challenges.
First, unlike photograph,opacity of brush strokes is a important feature of Chinese painting. It is desired to preserve this feature on bas-relief.
Second,unlike previous 2D image based methods focusing on photograph, a Chinese painting does not obey the rules of lighting and shading, which increase the difficulty to mimic the details on bas-relief surface.
Third, 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.

1.3 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 a 3D model,and some other methods based on a single photograph[9][10] [11][12] or a line drawing [14][15][16][17] 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[22], previous 2D image based methods may easily handle.
On the other hand,the style and philosophies of Chinese paintings have influenced other painting styles [22]. 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, 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.

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 automatic generate bas-relief from a Chinese painting.
In contrast with previous 2D image based methods, our algorithm is also the first attempt to generate bas-relief from opacity, and we demonstrate that in Chinese painting, opacity can preserve more details than intensity.
Accompanying with this method, we have proposed an algorithm for brush stroke extraction.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. And we demon-

strate the our method have a better performance in brush stroke extraction comparing with another notable work [18] (see Section 5.3).
Most Chinese paintings are typically sparse with each brush stroke drawn very purposefully[19], and such a feature can be found other painting styles,such as rosemailing painting[20]. So our method is also suitable for rosemailing painting, as shown in Section 7.1 and Section 5.4.

1.4 Document Structure

The organization of the document is as follows:
Chapter 1: Introduction. This section provides the background, the motivation, aim and challenges 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 and Future work. This section concludes the progress up-to-date and discusses possible future work.

Chapter 2

LITERATURE REVIEW

This chapter reviews notable bas-relief generation methods classified into two categories, methods based on 3D model and method based on 2D image, which are reviewed in section 2.1 and section 2.2 respectively. The structure of these bas-relief generation methods is shown in Figure 2.1. Following with section 2.3 which reviews notable brush stroke extraction methods and some relative works.

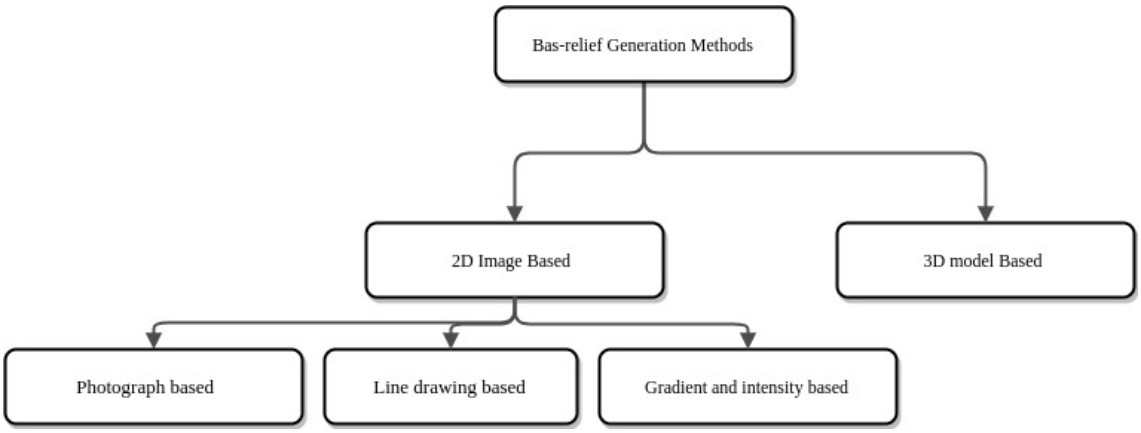


Figure 2.1: Bas-relief generation from brush strokes

2.1 Bas-relief Generation from 3D Model

A significant amount of methods has been considered bas-relief generation from 3D model,[6] firstly proposed a method which creates bas-relief models by linearly compressing (squeezing) the depth map of 3D models. This method fails to 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 Photograph Based Methods

Some methods generation bas-relief from a photograph [9][10] [11][12]. And the very importance part of these methods is generating bas-relief surface based on shape from shading(SFS) algorithm. SFS is a relative classic way for 3D shape recovery; see Zhang's survey [23]. 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[24], (1) orthographic SFS with a far light source [25]; (2) perspective SFS with a far light source[26]; (3) perspective SFS with a point light source at the optical center[24]; (4) a generic Hamiltonian. However, SFS is an ill-posed problem. The notable difficulty in SFS is the bas-relief ambiguity[13], 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[23] [27] [28] [2].

Unfortunately,these photograph based methods assume illumination and reflectance are known, and the image is formed from lighting and shading, which may work well for realistic photographs rather than paintings,especially for Chinese painting. Simply applying SFS method is not enough to generate bas-relief with acceptable quality .

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 line drawing. Kolomenkin et al.[14] 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 [15][16][17]. 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. [17] 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.[29] 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.
[18] 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 Extraction

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 extraction, 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 paint 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 paint colors. However the proposed method simply focus on layer decomposition not stroke extraction, how to extract the brush stroke remains a problem.
Li et al.[34] 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. In this research,they generated manually marked brush strokes using 10 example regions from van Gogh's paintings, and they define two parameters in order to evaluate the accuracy of extracted brush stroke. In section 5.3, we compared our method with this method.
Xu et al.[19] 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.And it requires professional artists to build a brush stroke library which makes it a challenging and time consuming process. 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[35] has been proved to be a very efficient way in stroke segmentation from scene text images[36][37], and the revised version has been widely used in stroke segmentation of handwritten characters[38].
The Most Stable Extremal Regions (MSERs) algorithm[35] was used for establishing correspondence in wide-baseline stereo.[39] 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.
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 [40] 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, our method can be separated into three steps: layer decomposition, brush stroke extraction, bas-relief generation.
Layer decomposition:
A given painting is firstly decomposed into a set of layers in terms of paint colors. In our decomposition, each layer represents a single-color coat of paint applied with varying opacity.
Brush stroke extraction:
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 on a black image to show the results of our brush stroke extraction, and each color(except black) represents the region of a single extracted brush stroke.
Bas-relief generation:
Thirdly, the depth map of each brush stroke is generated individually by the shape from shading techniques. 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 paint 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

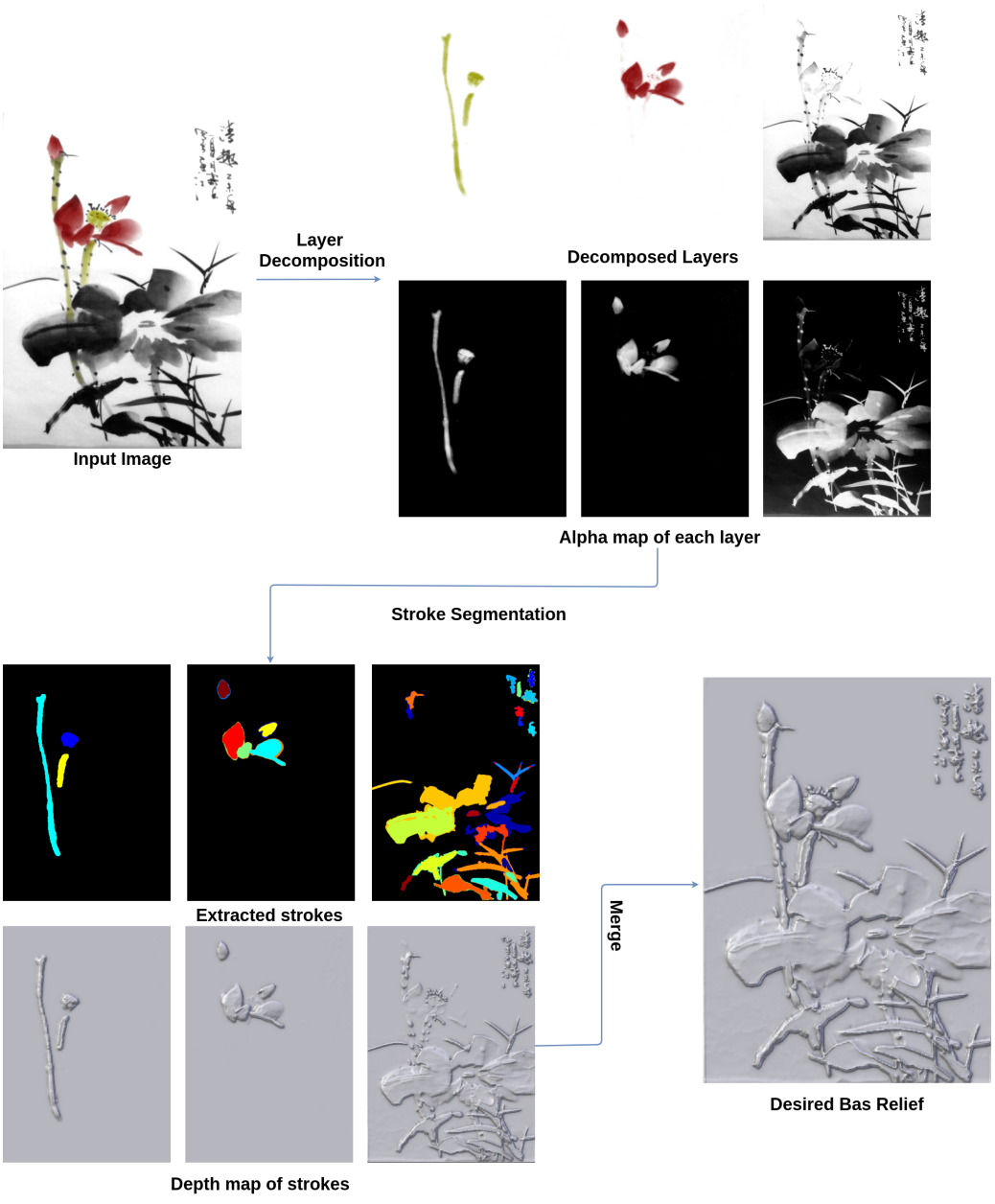


Figure 3.1: Overview of Proposed Method

layers. It is observed that most brush strokes in Chinese paintings the color and opacity change very little in the direction of the stroke [19]. So, we introduce the edge tangent flow (ETF) field and the coherent line [40] 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 [39] again for better extraction within one 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, 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 brush 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 [40] 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

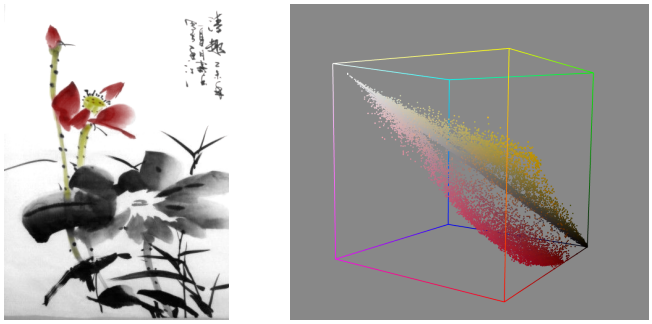


Figure 4.1: Geometry of input Chinese painting pixels in RGB-space

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.

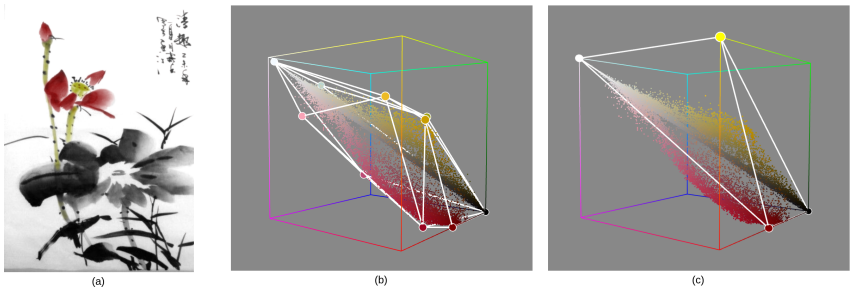


Figure 4.2: Convex hull of input image

(a)Input Chinese painting (b) Convex hull in RGB-space is complex due to rounding (c) The result of simplification algorithm

4.2 Tan et al.'s Layer Decomposition Scheme

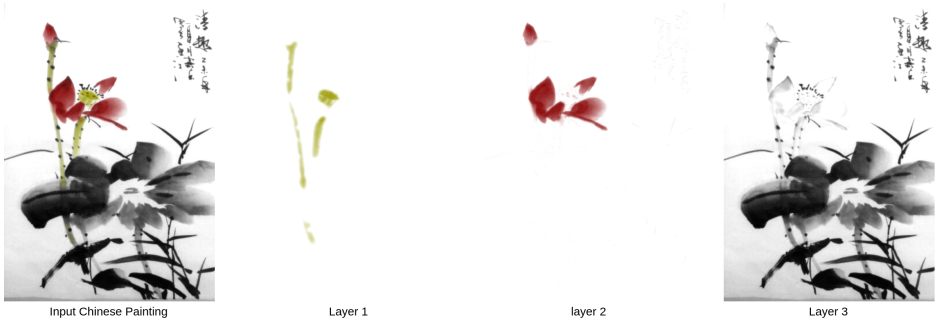


Figure 4.3: Decomposed layers

The standard Porter-Duff "A over B" compositing operation is very commonly used for color composition. The standard Porter-Duff "A over B" compositing operation in [41] 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,

$$\left(\frac{A}{B}\right)_{RGB} = \frac{\alpha_A A_{RGB} - (1-\alpha_A)\alpha_B B_{RGB}}{\left(\frac{A}{B}\right)_\alpha} \tag{4.1}$$

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 by taking input ordered RGB layer colors

to compute 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 [41],

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

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,

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

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,

$$E = \omega_{polynomial} E_{polynomial} + \omega_{opaque} E_{opaque} + \omega_{spatial} E_{spatial} \qquad (4.2)$$

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

4.3 Our Modified Layer Decomposition

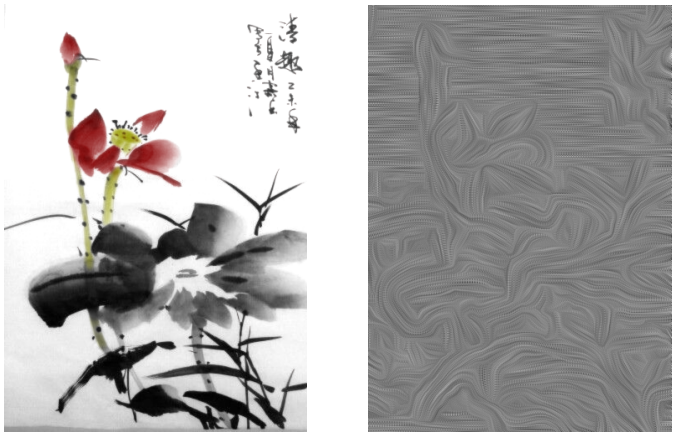


Figure 4.4: Edge Tangent Flow

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. It is observed that for most brush strokes in Chinese paintings,the color and opacity change very little in the direction of the stroke [19]. The nature question is how to detect direction of brush strokes. To detect direction of brush strokes and preserve the smoothness and completeness of brush strokes in layers, the coherent line drawing technique in [40] is introduced to Eq.4.2 ,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,

$$t^{new(x)} = \frac{1}{k} \sum_{y \in \Omega(x)} \varphi(x,y) t^{current}(y) \omega_s(x,y) \omega_m(x,y) \omega_d(x,y) \qquad (4.3)$$

$t(x)$ 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 4.5: Comparison of layer decomposition

Comparison of decomposed first layer before and after modification. Left shows the original result of Eq.4.2 , right shows the result by using E_{flow} instead of $E_{spatial}$ in Eq.4.2.

Involving ETF filed of Eq.4.3 in $E_{spatial}$, the smoothness penalty is rewritten as,

$$E_{flow} = \frac{1}{n} \sum_{i=1}^{n} \|t^{new}\| \left(\bigtriangledown_{\theta} \alpha_i\right)^2 \tag{4.4}$$

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.4.2 instead of $E_{spatial}$, the brush strokes become complete and smooth, which can be noted in Figure 4.5.

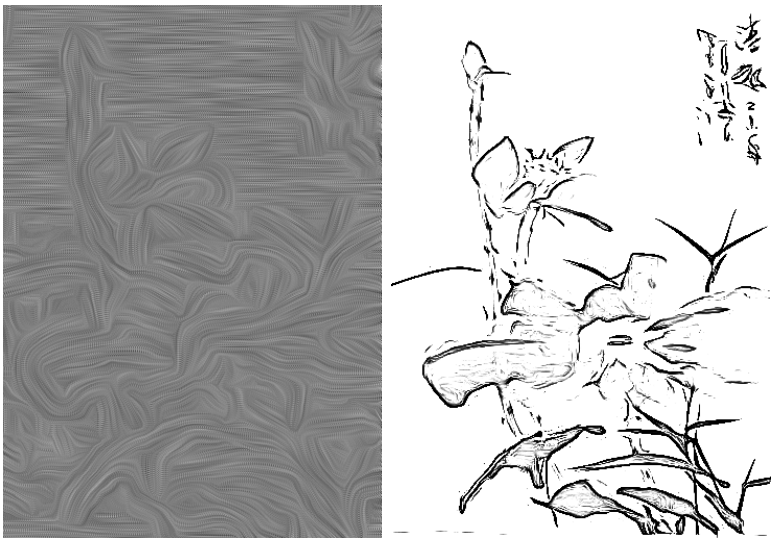


Figure 4.6: Edge Tangent Flow field and coherent lines of the painting.

Second, the coherent lines as the constraint of brush stroke edges are involved in layer decomposition of Eq.4.2. Herein, the coherent lines can be computed as follows.

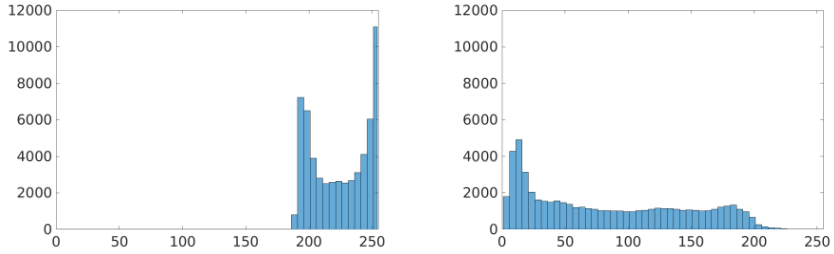
Figure 4.7: Comparison of layer decomposition before and after using E_{egde} in Eq.4.2;

Given a ETF field $t(x)$ to mimic the brush stroke direction, 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,see Figure 4.6. And we use the coherent lines to mimic the outlines of brush strokes.

(a) intensity map of layer2

(b) opacity map of layer2



(c) histogram of intensity map of layer2

(d) histogram of opacity map of layer2

Figure 4.8: Comparison between opacity and intensity of layer2

To preserve the brush 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 pixel

collection of coherent lines. Hence, the constraint term is defined by applying Laplacian operator to the opacity along the coherent lines,

$$E_{edge} = \|LY\|^2 \tag{4.5}$$

where all the opacity α_i are stacked in the vector Y, and L denotes the Laplacian 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 4.7 shows that the brush strokes become visible and complete after involving E_{edge} into Eq.4.2. Accordingly, the layer decomposition of Eq.4.2 is rewritten as,

$$E = \omega_{polynomial}E_{polynomial} + \omega_{opaque}E_{opaque} + \omega_{flow}E_{flow} + \omega_{edge}E_{edge} \tag{4.6}$$

We select the value of weights based on calculate the root-mean-square-error (RMSE) of the opacity of the coherent lines on each layer.By experiments of minimizing the RMSE, we set $\omega_{polynomial} = 365, \omega_{opaque} = 0.1, \omega_{flow} = 100, \omega_{edge} = 20$.
It is noteworthy that the opacity of the brush strokes is always independent of its color, and the intensity of image is a composition of color and opacity ,see Eq.4.1. So, naturally, opacity on decomposed layers reveals more details than intensity on the input image. Here, we use the decomposed second layer as example, see Figure 4.8.

4.4 Comparison and Evaluation of Layer Decomposition

To demonstrate our results can better preserve the smoothness and completeness of brush strokes, we perform the schemes of Eq.4.2 and Eq.4.6 separately on the same set of 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.4.6 is noticeably less than that by Eq.4.2. This means that the coherent lines have been embedded into the opacity of each layer.
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[42] and Vincent van Gogh's paintings[34] .

Table 1 Opacity RMSE of Coherent Lines on Layers

No.	Layers	Opacity RMSE of coherent lines	
		By Eq.4.2	By Eq.4.6
Chinese1 (548*732)	1	25.12	15.39
	2	8.89	5.92
	3	15.74	9.63
Chinese2 (472*784)	1	38.41	26.33
	2	21.28	16.37
	3	5.26	4.19
Chinese3 (387*651)	1	10.55	10.32
	2	15.27	11.84
	3	25.94	18.35
Chinese4 (560*812)	1	16.44	10.25
	2	20.54	15.04
	3	10.56	5.87
Chinese5 (516*875)	1	17.58	13.87
	2	19.78	8.72
	3	22.47	17.24
Chinese6 (426*915)	1	20.11	15.12
	2	18.14	10.24
	3	22.14	17.12
Chinese7 (516*475)	1	9.47	5.12
	2	18.41	12.99
	3	21.77	16.18
Rosemaling1 (400*522)	1	27.11	21.42
	2	17.52	14.58
	3	16.2	17.99
	4	19.24	11.51
Rosemaling2 (556*668)	1	15.44	12.84
	2	21.22	24.71
	3	25.72	19.95
	4	24.98	21.36
Van gogh1 (450*349)	1	10.18	28.11
	2	12.27	14.11
	3	22.34	13.41
	4	11.85	8.65
Van gogh2 (357*651)	1	15.89	14.25
	2	19.54	15.49
	3	17.21	12.98

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 decomposed layers by using 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 [39] 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 5.4.

The Maximally Stable Extremal Regions (MSERs) algorithm proposed in [39] is used 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 [39] 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,

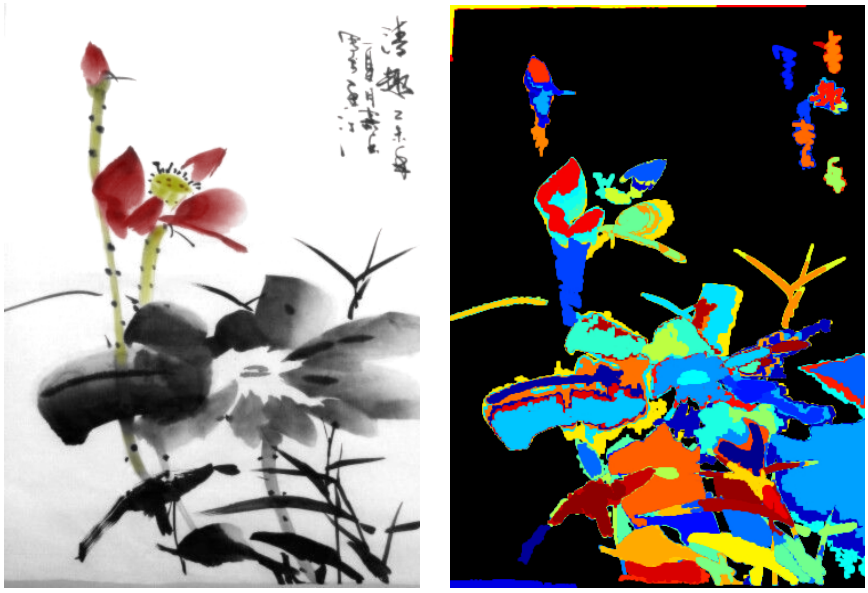
(1) The brush stroke with the intensity very close to the background;

(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 [40] 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



(a) input image (b) MSERs of the input image

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

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.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 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.

5.2 Modified MSERs Algorithm

In terms of the definition of the area change rate γ, the default MSERs algorithm 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. 4.6 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 fourth 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 defi-

nition 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. p,z,q are pixels belong to each collection. 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 5.2 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.

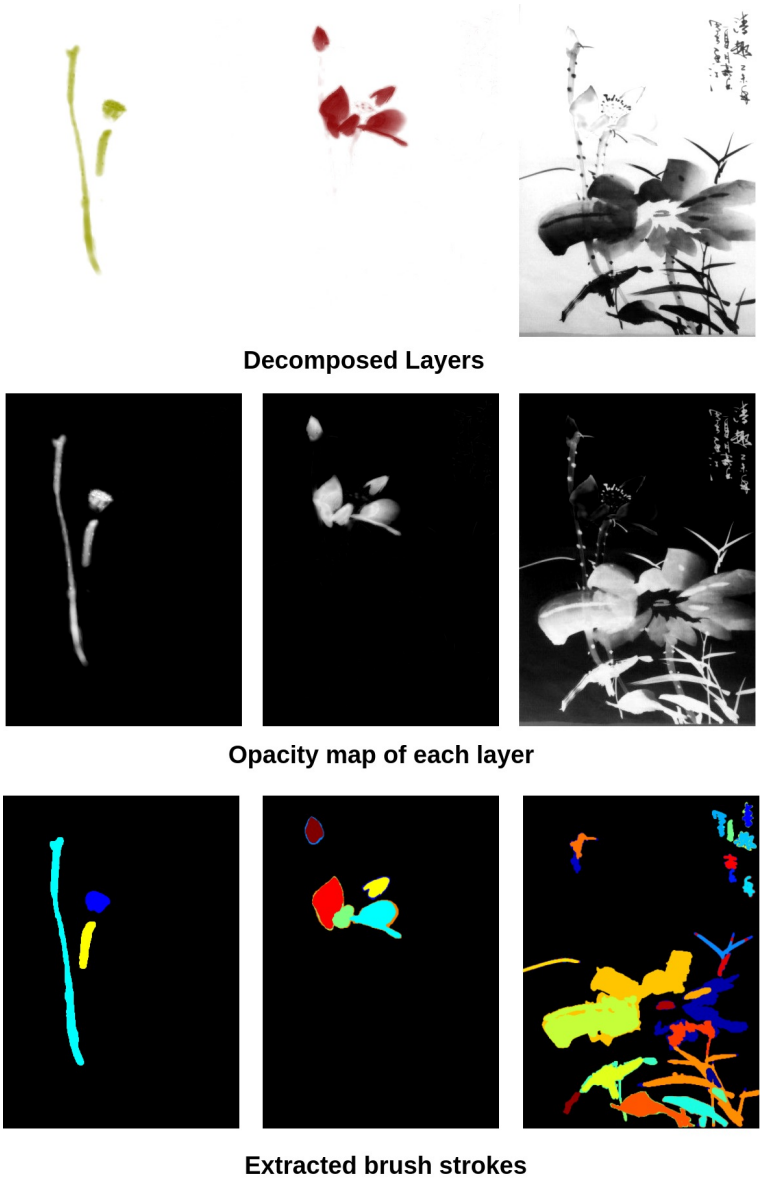
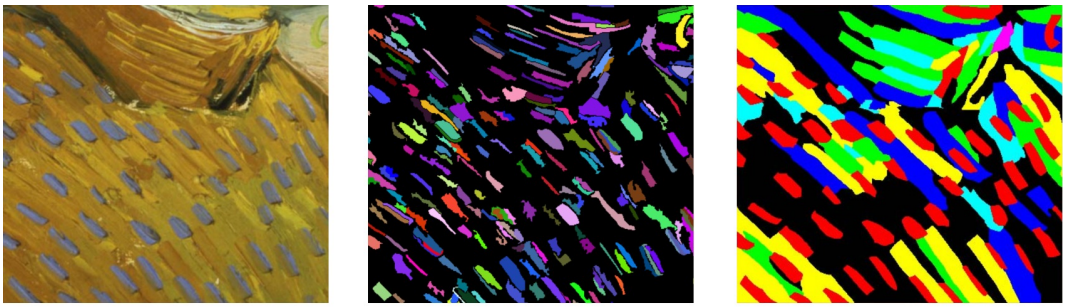


Figure 5.2: The opacity maps, and brush strokes extracted on three layers by the modified MSERs.

5.3 Comparison and Evaluation of Brush Stroke Extraction

To the best of our knowledge, there is lack of study on automatic brush strokes extraction. Li et al.[34] present a automatic brush stroke extraction method based on seed growing. To numerically evaluate their brush stroke extraction algorithm, they generated manually marked brush strokes using 10 example regions from van Gogh's paintings, see Figure 5.3.
Based on the manually marked examples, they define two parameters which can be measured in order to evaluate the accuracy of extracted brush stroke and,therefore, can be used to compare between their method and ours: valid rate and detection rate.
Valid rate:the percentage of valid automatically extracted brush strokes.
Detection rate: The percentage of detected manual brush strokes .
Even through our brush stroke extraction algorithm is specifically not designed for van Gogh's paintings, for fair comparison, we use the van Gogh's painting given in their research, as shown in Figure 5.4.
As shown in Table 2,the valid rate and detection rate of our method is noticeably higher than Li et al.'s method [34] and default MSERs algorithm (The paint ID is given by Li et al.'s work).



(a) Input painting region (b) Brush stroke extracted by Li et al's method (c) Manually marked brush strokes

Figure 5.3: Brush strokes extraction in Li et al.'s work

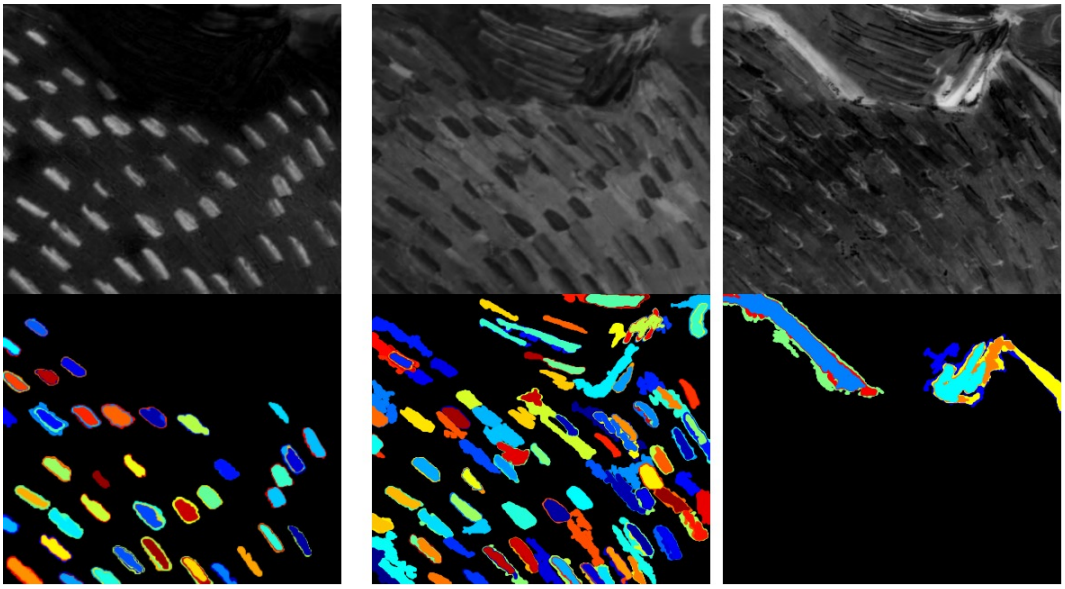


Figure 5.4: Brush strokes extraction by our method; the upper row shows the opacity maps of each layer

Table 2 Brushstrokes Extraction Evaluation and Comparison

	Valid Rate (%)			Detection Rate (%)		
Painting ID	**Li et al.'s method**	**Default MSERs**	**Our method**	**Li et al.'s method**	**Default MSERs**	**Our method**
F218	42.7	79.8	88.1	21.6	33.7	48.5
F248a	75.4	79.4	90.2	78.8	82.1	88.1
F297	57.9	67.2	75.3	52.0	61.0	75.1
F374	58.2	67.9	75.9	63.2	60.8	82.4
F386	73.7	64.1	82.4	68.4	59.2	90.2
F415	46.9	50.3	58.1	60.0	80.1	92.1
F518	60.7	55.7	71.2	75.2	86.1	78.9
F538	49.0	58.4	84.2	44.9	61.4	81.3
F572	83.9	68.9	82.4	65.6	71.2	87.0
F652	50.0	60.1	75.8	72.5	60.1	67.3
Average	59.8	65.2	78.4	60.2	65.6	77.1

As shown in Table 2, our method extracts brush strokes with higher accuracy.

5.4 Image editing based on brush strokes

Once the brush strokes are extracted,we can simply recoloring strokes by assign a RGB value to the brush stroke while keep its opacity value. Figure 5.6 a and 5.6 b shows recoloring brush strokes on three paintings respectively. As the brush strokes have been extracted, it is easy to separately recolor one or more strokes with different colors.

Figure 5.6 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 brush strokes here. The occluded regions of the objects are visible due to the transparency of the brush strokes. Unlike the traditional image synthesis approaches, our implementation works on the strokes, which both guarantees seamless and keep the opacity of brush strokes on the painting, see Figure 5.5.

(a) Simply paste object into the painting

(b) Insert object between strokes

Figure 5.5: Insert object between brush strokes

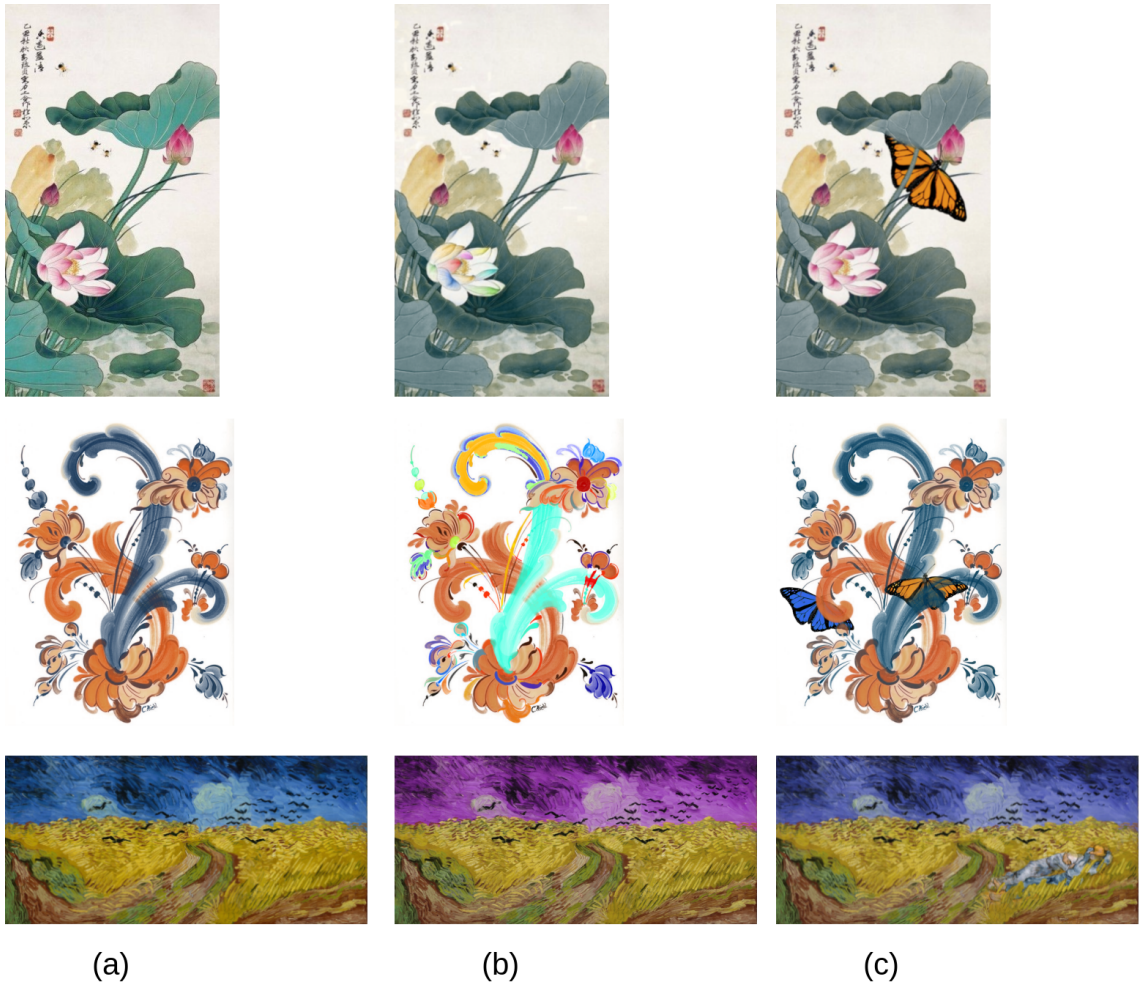


Figure 5.6: Image editing based on extracted brush strokes

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 brush strokes in one layer.Column (c) shows inserting objects into these 3 paintings, in which the objects are opaque and are inserted in between brush strokes.

Chapter 6

BAS-RELIEF GENERATION

Since the brush strokes are 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 we use a depth map to represent a bas-relief model,which is commonly used for bas-relief generation methods. In this Chapter, we demonstrate how to generate bas-relief from brush strokes. 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 [26] algorithm on the segmented brush strokes. The brightness equation used in the SFS algorithm is expressed as,

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

$I(x)$ is the intensity at pixel x, $h(x)$ is the depth at pixel x. It can be noted that for higher intensity I, change of depth h is smaller. Some brush strokes are painted by colors with high intensity. As a result, if the shape from shading algorithm is performed on intensity, the resulting stroke models will become flat and lack of hierarchy. 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 :

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

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

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

where Δ is a positive displacement. This modification may make the surface inflated. By using fast marching algorithm [43] to solve this equation, we can generate 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. Then by applying our algorithm we can generate bas-reliefs(depth maps) from each one of the brush strokes on this layers.By generating depth map from all brush strokes,and merge them together, we automatically output a bas-relief from the input Chinese painting.
To differentiate the bas-relief generated form a single stroke and the bas-relief generated from the whole painting, 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 [19] ,in our approach, a bas-relief can be regarded as a union of stroke-relief.Since the stroke-reliefs are independent of each other, the order and shape may be redefined by the users. It is suitable for further editing in bas-relief design.

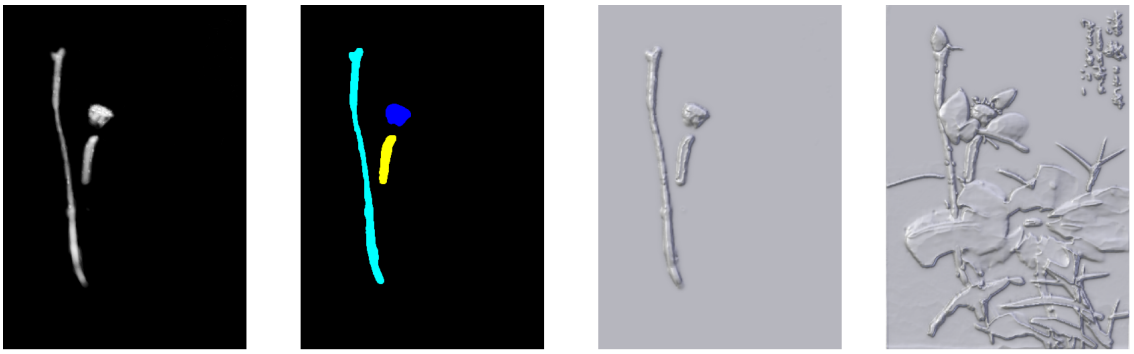


Figure 6.1: Bas-relief generation from brush strokes

Figure 6.2 and Figures in section 7.1 show our bas-relief results generated for various Chinese paintings.

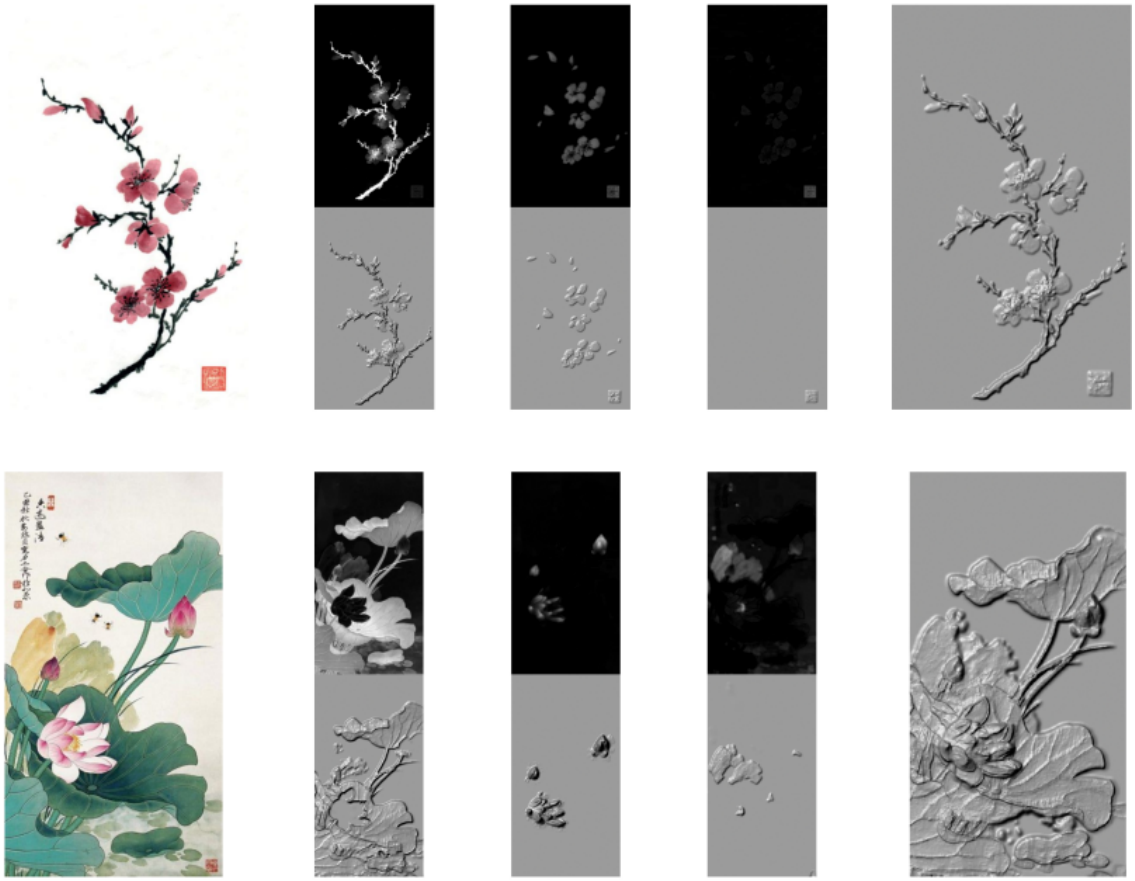


Figure 6.2: Bas-relief generation from Chinese paintings

6.1 Comparisons of Bas-relief Generation

Most of other 2D images based methods is unsuited for our aim due to some simply reasons, which is explained in section 1.1.2 and section 2.2.
Since our generation is based on opacity of brush strokes and each stroke occupies a 2D region on canvas,in this section, we compares our bas-relief generation method against another alternative notable 2D image based bas-relief generation method [9].
Zeng et al.'s work [9] present region-based algorithm for bas-relief generation.At first,their method segments input image into 2D regions and determines the layer of the regions. Then, they reconstruct relief from each layer. And they demonstrate this algorithm works well for a range of input photos, including human faces, flowers and animals.
In their method, to extract regions from the image, the feature lines are de-

tected at first. Next, seed points are derived from these lines, and regions are found from them using a region growing process. The region extraction in their method is quite similar with our brush stroke extraction process. For Chinese painting, a brush stroke can be considered as a region on canvas. So we compare our method against Zeng et al.'s method [9] based on three factors mentioned in Section 1.1.2 : depth information, silhouettes and edges,fine details.

Depth information: In the Zeng et al.'s work [9] , regions determine the depth information, so regions have to be faithfully extracted. The first step of extract regions is detect region outlines by line detection on intensity map. Naturally, the line detection fails in 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;

So, their algorithm would fail at the first step.

In our algorithm, the above mentioned problems is avoided by layer decomposition and stroke extraction , see Section 5. By extracting brush stroke faithfully, we follow the manner of Chinese painting, generate bas-relief from all brush strokes.

Silhouettes and Edges: As mention above, the region-based algorithm fails to find the outlines of brush strokes, namely, it can not separate brush strokes from the background or generate a reasonable contrast between the foreground objects and the background as a starting establishment for interpreting heights. On the other hand, since our method apply layer decomposition at first, foreground(brush strokes) is easily separated from background.

Fine Details: Region-based algorithm performs shape from shading on the intensity of the image.As we mentioned,for brush strokes painted by bright color, the generated bas-relief would be flat and features can not be well preserved. In our algorithm, we use opacity to generate the bas-relief, which maintains the details of brush strokes better, see Figure 4.8.

Chapter 7

RESULTS

Implementation We modified the published codes of the layer decomposition and MSERs, which are available on GitHub (https://github.com/CraGL/Decompose-Single-Image-Into-Layers; https://github.com/idiap/mser) ,the differences between our algorithm and the available codes are explained in Section 4.3 and 5.2. Our algorithms were written in Python and vectorized using NumPy/SciPy. In our implementation, the parameters in Layer decomposition, ETF field, and MSERs are set the default values as in the original codes.

Performance We performed the bas-relief generation on a variety Chinese paintings, and some Rosemaling paintings. We also performed our brush strokes extraction algorithm on some van Gogh's paintings, as described in Section 5.3. All the paintings are from the internet. All the tests were performed on a 6-core of 3.33 GHz Intel Core Xeon CPU with memory of 32 GB(RAM).

Table 3 further shows the running time of the proposed approach. Compared to the performance in [33] and [44], there is no distinct difference. Additionally, the shape from shading algorithm does not consume much time since it depends on the segmented stroke regions. Our implementation is not multithreaded.

Comparisons As illustrated in section 3, our method can be separated into three steps:layer decomposition, brush stroke extraction, bas-relief generation. We compared our work in each step with other alternative methods.

For layer decomposition, we compared the our algorithm with the Tan et al's work [33] in section 4.4, and we demonstrate our modified algorithm can better preserve the smoothness and completeness of brush storkes in layers.

For stroke extraction, we compared our work with Li et al.'s work [34] in section 5.3, and demonstrate that our method have a better performance in brush stroke extraction.

For bas-relief generation,we compared our work with Zeng et al.'s work [9] in section 6.1, and we demonstrate our algorithm has better performance based on three factors mentioned in Section 1.1.2 : depth information, silhouettes and edges,fine details.

Limitations.

We discuss the limitations of this work as related to different steps in the method: 1) Layer decomposition: The paint colors based on building and simplify the convex hull of input image in RGB space, sometimes, a paint color can be inside of the convex hull which can not be selected. With wrongly picked paint colors, the layer decomposition may not preserve or separate the brush strokes, see Figure 7.1 and Figure 7.2.

2) Brush stroke extraction: some images can be too complex for proper segmentation, e.g., brush strokes highly blended with painting colors; hence, we can not form meaningful regions to generate bas-relief, as shown in Figure 7.3.

Figure 7.1: The green color on the headdress is wrongly separated into second layer

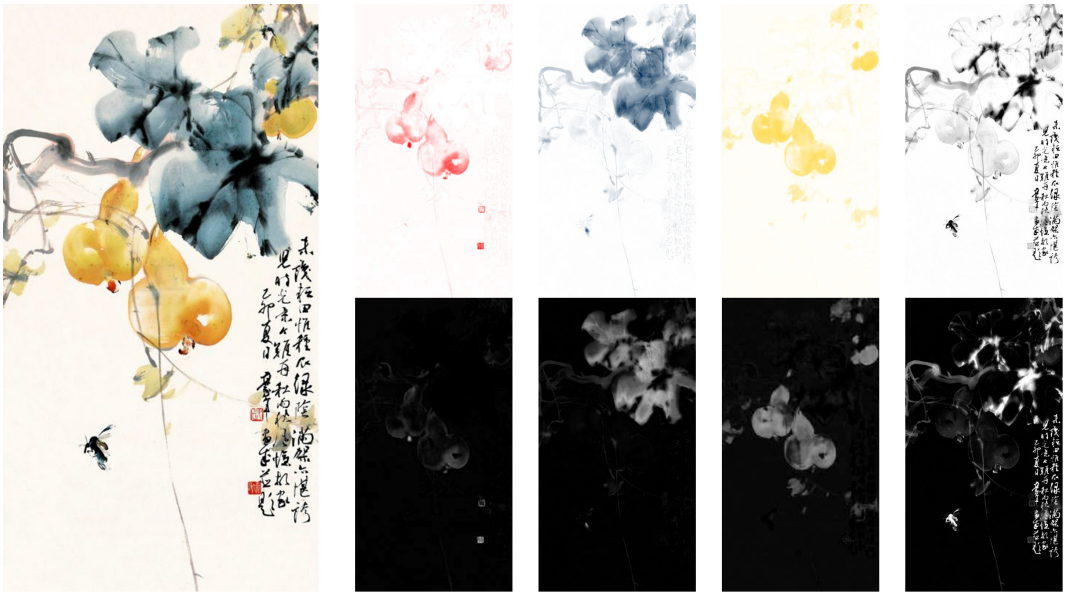


Figure 7.2: The color of gourd is separated into two layers

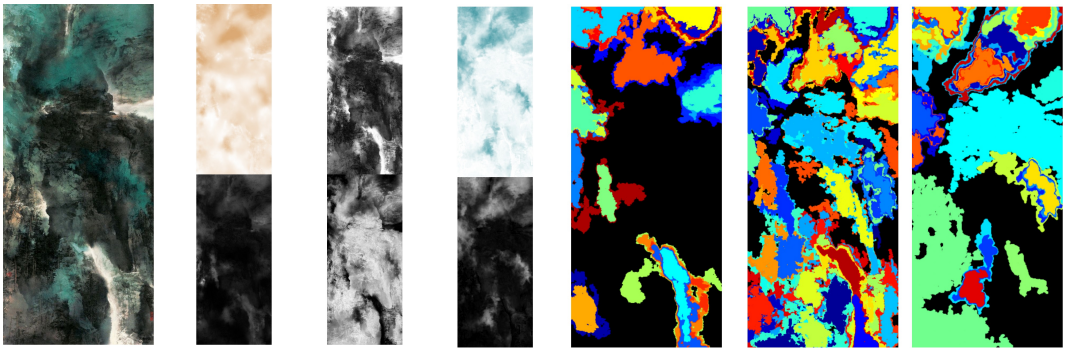


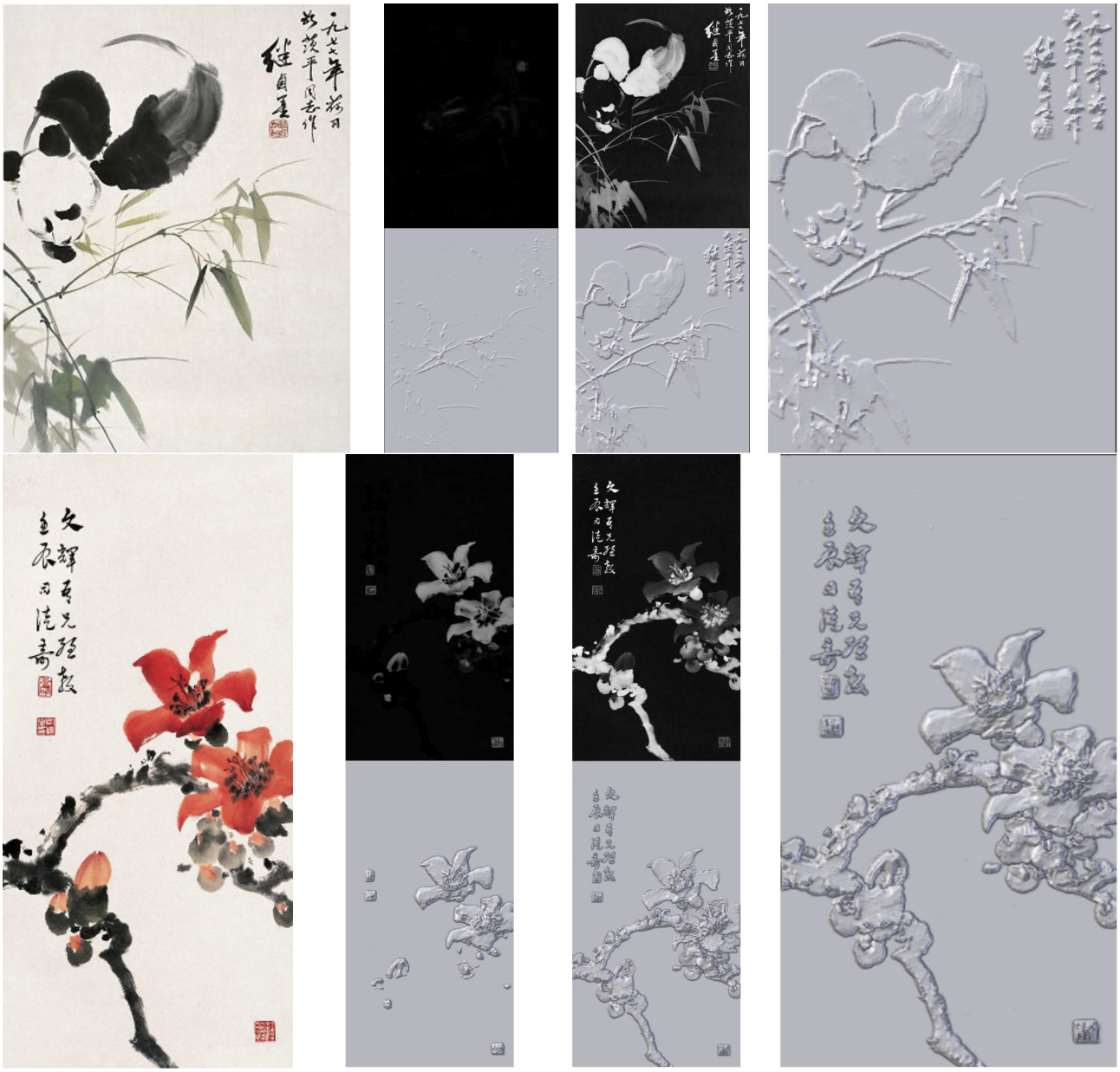
Figure 7.3: Brush strokes highly blended with painting colors

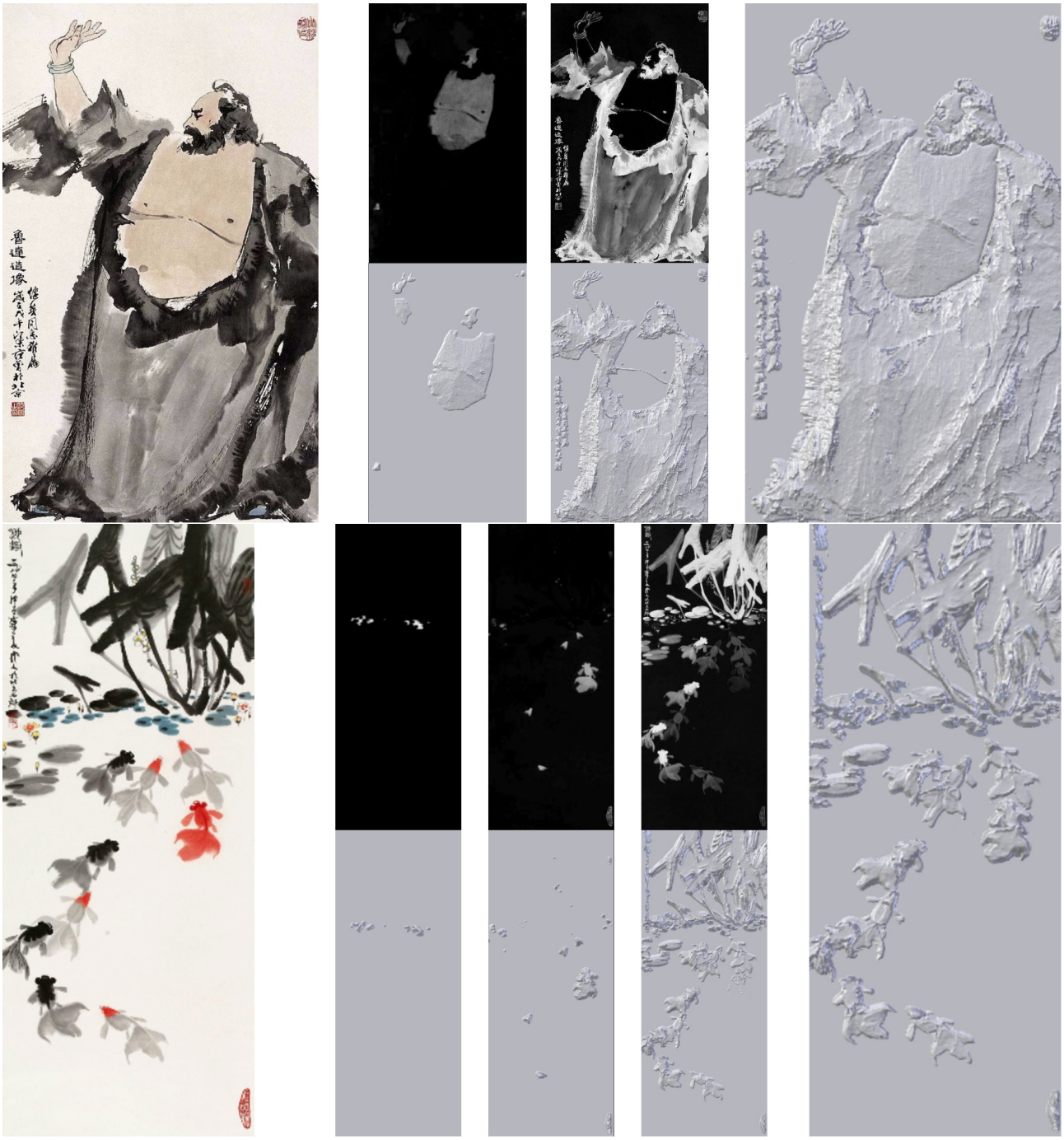
Table 3 Performance

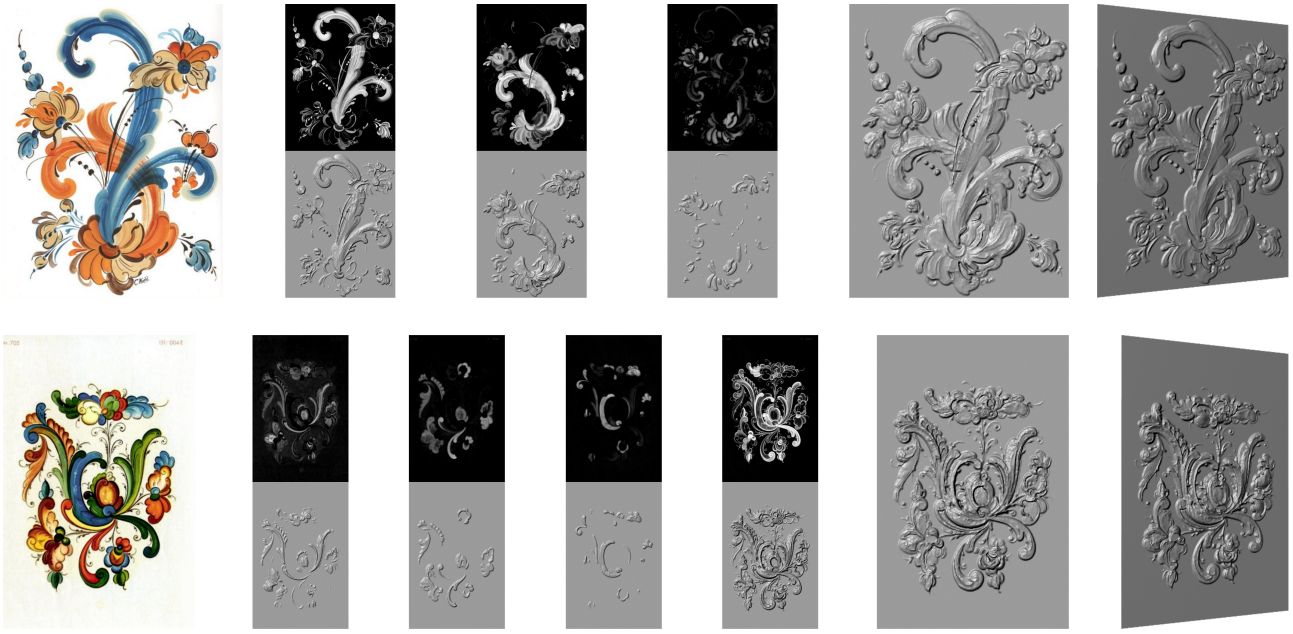
No.	Stroke Number	Layers	Opacity RMSE of coherent lines		Runtime of Layer (sec)	Runtime of MSERs(sec)
			By Eq.4.2	By Eq.4.6		
Chinese1 (548*732)	110	1	25.12	15.39	162.21	0.85
		2	8.89	5.92		
		3	15.74	9.63		
Chinese2 (472*784)	38	1	38.41	26.33	70.54	0.32
		2	21.28	16.37		
		3	5.26	4.19		
Chinese3 (387*651)	201	1	10.55	10.32	314.98	1.51
		2	15.27	11.84		
		3	25.94	18.35		
Chinese4 (560*812)	258	1	16.44	10.25	387.11	0.85
		2	20.54	15.04		
		3	10.56	5.87		
Chinese5 (516*875)	292	1	17.58	13.87	311.25	0.64
		2	19.78	8.72		
		3	22.47	17.24		
Chinese6 (426*915)	358	1	20.11	15.12	251.88	0.98
		2	18.14	10.24		
		3	22.14	17.12		
Chinese7 (516*475)	487	1	9.47	5.12	271.35	0.74
		2	18.41	12.99		
		3	21.77	16.18		
Rosemaling1 (400*522)	114	1	27.11	21.42	164.87	0.75
		2	17.52	14.58		
		3	16.2	17.99		
		4	19.24	11.51		
Rosemaling2 (556*668)	220	1	15.44	12.84	157.32	0.45
		2	21.22	24.71		
		3	25.72	19.95		
		4	24.98	21.36		
Van gogh1 (450*349)	1668	1	10.18	28.11	384.68	0.51
		2	12.27	14.11		
		3	22.34	13.41		
		4	11.85	8.65		
Van gogh2 (357*651)	1501	1	15.89	14.25	354.52	3.74
		2	19.54	15.49		
		3	17.21	12.98		

7.1 More Results

More bas-relief generation based on different paintings







Chapter 8

Conclusions and Future Work

8.1 Conclusions

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 automatically.
Based on layer decomposition , our approach can effectively extract brush strokes from 2D paintings and generate the individual depth maps for bas-relief. We show the superiority of our results by comparing it with alternative works in each step.

In summary,our contributions include:
(1) Extraction of brush strokes. We develop a novel method to extract brush strokes based on paint colors 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 and has higher accuracy.
(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, we preserve the feature of the input painting well.
(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.
(4)Our brush stroke extraction technique has other potential uses. We demonstrate the utility of the decomposed brush strokes for image editing. See section 5.4.
Our experiments show that our method is able to produce convincing bas-reliefs from a variety of Chinese brush paintings (with human, animal, flower,etc.)

and even some other suitable styles including rosemaling paintings.

8.2 Future Work

Although the proposed algorithm is already capable to generate a bas-relief from a single Chinese brush painting automatically, there is still further work to do, including refinement of our method based on the limitations,and use it in other applications.

8.2.1 Refine Current Algorithm

One future work is to further improve the performance of current algorithm. As we mentioned in section 7, there are two main limitation: (1) With wrongly picked paint colors, the layer decomposition may not preserve or separate the brush strokes. (2) When brush strokes highly blended with multiple painting colors, it fails to be extracted faithfully by our current method.
We plan to investigate methods to automate the estimation of better color selection from the image contents, e.g., by analyzing the local features. Second, we would study how color composition equation would influence our brush stroke extraction, and refine our method based on it.

8.2.2 Application

3D painting system
The concept of 3D painting system was introduced for more than two decades ago. Hanrahan [46] 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. [47], 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 [48], 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 [49] 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.

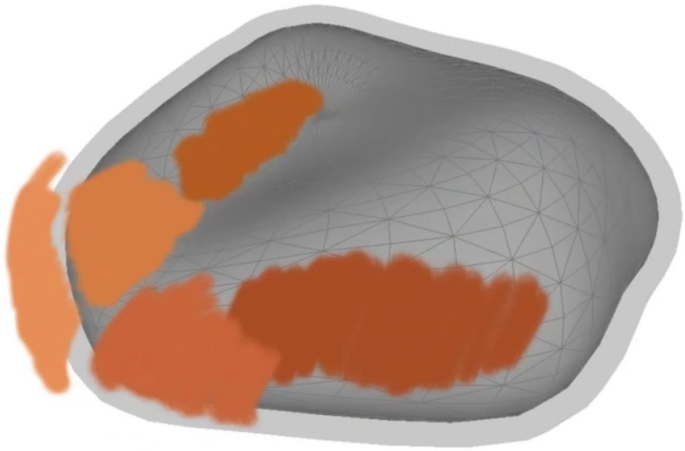


Figure 8.1: 3D brush painting on surface level set

More recently, Overcoat [50] 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 8.1. These painting systems are focusing on how to paint on surface of known models. In CanvoX [45] extended painting surface to volumetric painting, with GPU-based Octree [51].

With the rapid development of VR systems, it is natural to enable artists to paint in 3D space. In CavePainting [52] 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 [45]. For these painting system, quad strips are rendered as painting strokes, which are fully based on user input.

Application in 3D Painting

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 make 3D painting more and more intuitive for 2D painters. We aim to build a system which enable artist to transfer a given 2D Chinese 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 Chinese painting.
We address this problem into four main steps:
First, segment strokes from the given 2D Chinese painting. Since in 3D 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 [45].
Fourth, 3D strokes placement, based on the layer order and user inputs, we place and blend 3D strokes in 3D space.

8.3 Proposed timescale for the work:

Refine the first step of layer decomposition, with the coherent edge of the given 2D brush painting, iteratively calculate the paint colors. (6 weeks)
Stroke segmentation into a hierarchical structure, currently, hierarchical clustering are in consideration. (4 weeks)
Design the user interface for refine the stroke segmentation. (4 weeks)
With given segmented strokes, generate corresponding mesh, with detail and style preserved. (4 weeks)
Design the interface for user input, in which high relief can be placed naturally in 3D space while maintaining an acceptable level of controllability. (4 weeks)
Optimization. (4 weeks)

Table 8.1: Schedule of Future Research Plan

Research Activity	**Target Completion Date**
Refine current algorithm	11/12/2017
Design the user interface for refine the stroke segmentation.	01/3/2018
3D stroke analogy design and 3D stroke placement design .	01/04/2018
Optimization.	01/05/2018
Write final Thesis for PhD graduation	01/10/2018

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] Tai-Pang Wu, Jian Sun, Chi-Keung Tang, and Heung-Yeung Shum. Interactive normal reconstruction from a single image. In *ACM Transactions on Graphics (TOG)*, volume 27, page 119. ACM, 2008.

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

[14] 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.

[15] 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.

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

[17] 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.

[18] 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.

[19] 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.

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

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

[22] 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.

[23] 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.

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

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

[26] 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.

[27] 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.

[28] 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.

[29] 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.

[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] 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.

[35] 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.

[36] 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.

[37] 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.

[38] 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.

[39] 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.

[40] 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.

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

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

[43] James Albert Sethian. *Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science*, volume 3. Cambridge university press, 1999.

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

[45] 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.

[46] 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.

[47] 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.

[48] 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.

[49] 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.

[50] 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.

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

[52] 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.

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

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

[55] 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.