

BOURNEMOUTH UNIVERSITY

TRANSFER DOCUMENT

From Brush Painting to Bas-relief

Author:
Yunfei FU

First Supervisor:
Dr. Hongchuan YU
Second Supervisor:
Pro. Jian Jun ZHANG

September 20, 2017

Contents

1	INTRODUCTION	3
2	LAYER DECOMPOSITION	6
2.1	Identify Paint Colors	6
2.2	Layer Decomposition Scheme	7
2.2.1	Modified Layer Decomposition	8

List of Figures

2.1	Geometry of input image pixels in RGB-space	6
2.2	Convex hull of input image	7
2.3	Edge Tangent Flow	8
2.4	Coherent line details	9
2.5	Comparison of layer decomposition	10
2.6	Edge Tangent Flow field and coherent lines of a Rosemaling painting. It contains lots of C and S strokes.	10
2.7	Comparison of layer decomposition before and after using E_{edge} in Eq 2.1;	11
2.9	Comparison between transparency and intensity of layer1	12
2.10	Comparison between transparency and intensity of layer1	14

Chapter 1

INTRODUCTION

Relief is a kind of sculpture in which 3D models are carved into a relatively flat surface. In essence, it creates a bridge between a full 3D sculpture and a 2D painting. As an artistic form, relief spans the continuum between a 2D painting and a full 3D sculpture[1]. On this spectrum, alto-relievo (high relief) is closest to full 3D, whereas flatter artworks are described as basso-relievo (low relief, and also called bas-relief). Among all the sculpture forms, bas-relief is arguably the closest to 2D paintings, as claimed by[2].

More recently, most existing bas-relief production methods focus on compressing 3D scenes/models into surfaces with a small depth[1]and[2]. This approach requires a 3D model as a starting point.

Another option is to generate bas-reliefs directly from 2D images. There have been some image based bas-relief production approaches available in[3][4] and [5]. These approaches almost follow the “bas-relief ambiguity”[6], that is, roughly speaking, from a frontal view the sculpture looks like a full 3D object while a side-view reveals the disproportional depth.

While most image based method are focusing on general photograph, which unsuited for specific problems for brush paintings, such as overlapping and stroke transparency. A another clear shortcoming is current image based methods can't take the artistic intent into account, as all what they do is to inferring the height information from the image. Concerning reproduction or modifying of an artistic painting, it is crucial that the style of the originals is preserved, which is not considered in existing image-based methods.

However, little is done in the area of bas-relief generation from artistic paintings, as maintaining the styles of the brush paintings proves much trickier than simply manipulating the height of the contour lines. In the case of bas-reliefs, although there is no 3D model available, pseudo 3D effect reflecting the style and subtlety is crucial in preserving the artistic essence.

The aim of our research is to provide the bas-relief sculptors with a new tool allowing them to convert and recompose existing brush paintings to bas-reliefs. We also argue that because traditional paintings are produced with individual strokes, ‘3D bas-relief strokes’ will enable them to ‘paint/sculpt’ a bas-relief naturally, especially if they wanted to quickly convert an existing painting into a relief. With the commonly and cheaply available 3D printing facilities, there is a growing trend in the need of bas-relief art products.

A brush painting can be regarded as the union of a set of hypothetical strokes by a brush [7]. Differing from the other bas-relief generation methods, our method will honor

this very feature by constructing the brush strokes individually as 3D geometric entities. This however demands to conquer several challenges. First, each brush stroke covers a region on the canvas and they may overlap each other, some quite heavily in a painting. To make sure the information is retained, every stroke has to be faithfully extracted. Second, spatial occlusion has to be dealt with, since artists are used to depicting it through controlling the transparency of strokes as one of the art elements. Third, as an artistic tool, the generated bas-relief should be further editable allowing the artist to rearrange, tweaking and reshaping the extracted strokes.

The shape, color and opacity of a stroke vary due to the shape and firmness of the brush as well as the forces the artist imposes. Although these variables add the complexity to stroke extraction, stylized strokes often follow distinct patterns. For example, Rosemailing paintings, a typical example of brush painting popular in North Europe, is a traditional form of decorative folk art that originated in the rural valleys of Norway. The Rosemailing designs use C and S strokes, feature scroll, flowing lines, floral designs, and both subtle and vibrant colors. The brush strokes may further be viewed as graphical objects which are meaningful with respect to the objects the painting portraits. Moreover, each stroke is clearly visible due to both subtle and vibrant colors. The similar properties may be found in some Chinese brush paints.

To extract the strokes from a brush painting, we need to identify and segment the overlapped strokes. We will then generate the depth map for every stroke separately using the shape from shading (SFS) technique on the opacity. All the strokes are finally merged together to yield the resulting bas-relief with the original 2D painting preserved. Our contributions include,

- (1) Extraction of brush strokes. We develop a novel method to extract brush strokes from input paintings with palette analysis and decomposed layers.
- (2) 3D modeling of brush strokes. We develop a novel method which may entirely construct every stroke in 3D based on the opacity of paintings.
- (3) Recomposition in bas-relief design. Artists may redefine the brush strokes' order and shapes by sketches, which enable recomposition in bas-relief design, making it a useful tool for sculpture artists.

Overview As showed in Figure ??, the approach performs in three steps:

First, based on the point cloud of the input image in RGB space (Figure 2.1), we select the palette colors of the input brush painting, and based on the palette colors we decompose a input image into different layers with transparency,

Second, the original painting is decomposed into element brush strokes by applying modified MSERs segmentation. Based on our observation, alpha map is more suitable for our MSERs segmentation, so the segmentation is based on the transparency. Third, based on the segmented brush strokes, Shape from Shading algorithm has been applied to generate the depth map of each stroke.

Finally, based on those depth maps of strokes, we can edit the generated bas relief. For editing, we have already generated the 3D proxies for each MSER region, namely, the extracted 2D brush region, so, we can change the depth of specific stroke on bas relief, we can also stitch them on each other based on the user input and change the shape of a certain stroke with given indicated skeleton. By doing so, we fulfill the request of recomposition in bas-relief design. The resulting 3D proxies of brush strokes are sufficient to evoke the impression of the consistent 3D shapes, so that they may be further edited in

3D space. Currently, our research focus on brush paintings with relatively sparse strokes. Experiments show that our method can effectively generate digital bas-reliefs for a range of input images, including some Chinese paintings and rose-mailing paintings. We also demonstrate the utility of the resulting decompositions for image recoloring and image object insertion and animation.

Chapter 2

LAYER DECOMPOSITION

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

In general, each layer represents one coat of a painting with single color that is applied with varying opacity throughout the input painting. Wrong layer decomposition may cut one stroke into different layers. It is crucial to preserve the completeness and smoothness of the brush strokes in layer decomposition. To this end, we modify the layer decomposition algorithm in by involving the coherent lines [8] in our implementation. In the following we first address the layer decomposition algorithm briefly and then discuss our modification.

2.1 Identify Paint Colors

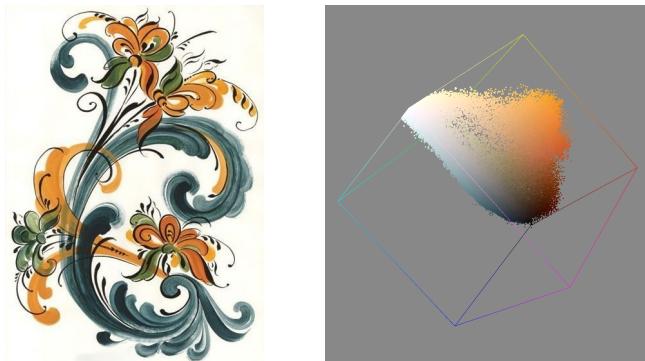


Figure 2.1: Geometry of input image pixels in RGB-space

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 2.1. And base on such idea, we can represent each pixel color based on painted color:

$$p = \sum \omega_i c_i$$

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 [9]. In which a convex hull of the colors in RGB space should be computed, while every vertex 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 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 2.2.

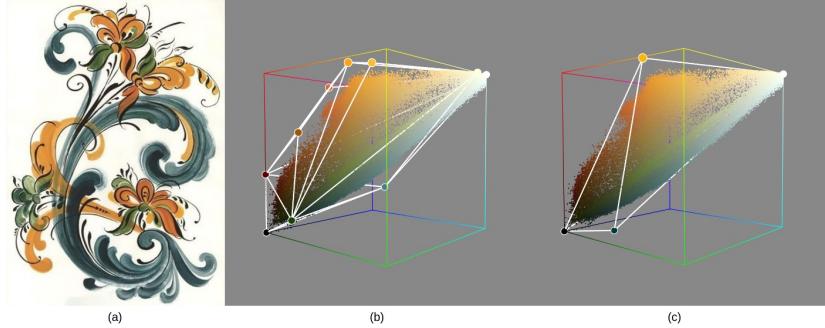


Figure 2.2: Convex hull of input image

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

2.2 Layer Decomposition Scheme

The “A over B” compositing and blend mode in [10] is described that when the pixel A with color A_{RGB} and translucency α_A is placed over the pixel B with color B_{RGB} and translucency α_B , the observed color is,

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

where

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

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

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

where C_i denotes the i -th layer's color, α_i 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,

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

where $\nabla \alpha_i$ 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,

$$E = \omega_{polynomial} E_{polynomial} + \omega_{opaque} E_{opaque} + \omega_{spatial} E_{spatial} \quad (2.1)$$

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

2.2.1 Modified Layer Decomposition



Figure 2.3: Edge Tangent Flow

As we can see in Figure ??, the layer decomposed, to enhance the smoothness and completeness of strokes, the coherent line drawing technique in [8] is introduced to Eq 2.1, which is a flow-guided anisotropic filtering framework. Figure 2.3 shows the edge tangent flow (ETF) field of a Rosemaling painting. First, the ETF field is involved in $E_{spatial}$. The ETF field is defined as,

$$t^{new(x)} = \frac{1}{k} \sum_{y \in \Omega(x)} \varphi(x, y) t^{current}(y) \omega_s(x, y) \omega_m(x, y) \omega_d(x, y) \quad (2.2)$$

As showed in Figure 2.4, $t(x)$ denotes the normalized tangent vector at pixel x , $\Omega(x)$ denotes the neighborhood of the pixel x , and k is the term of vector normalization. The

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

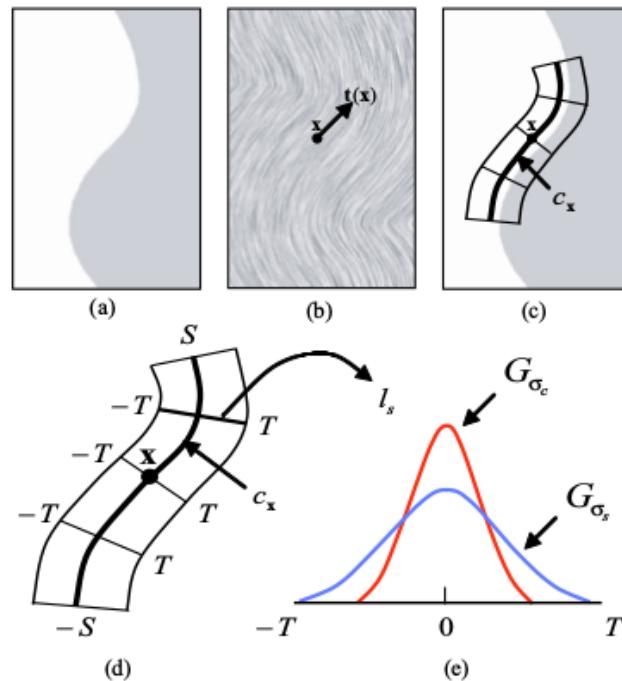


Figure 2.4: Coherent line details

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

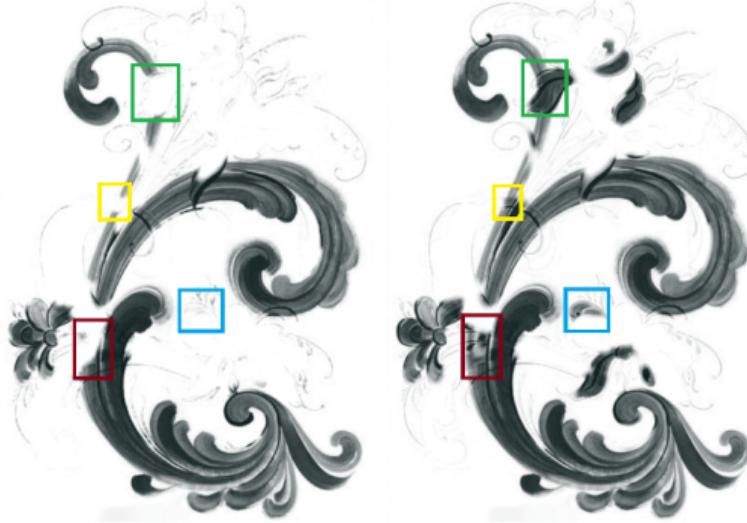


Figure 2.5: Comparison of layer decomposition

Comparison of layer decomposition before and after modification at the 2nd layers. left and right show the results by using E_{flow} instead of $E_{spatial}$ in Eq 2.1;

Involving ETF filed of Eq 2.2 in $E_{spatial}$, the smoothness penalty is rewritten as,

$$E_{flow} = \frac{1}{n} \sum_{i=1}^n \|t^{new}\| (\nabla_\theta \alpha_i)^2 \quad (2.3)$$

where θ denotes the direction of t^{new} , and $\nabla_\theta \alpha_i$ 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 2.1 instead of $E_{spatial}$, the strokes become complete and smooth, which can be noted in Figure 2.5.

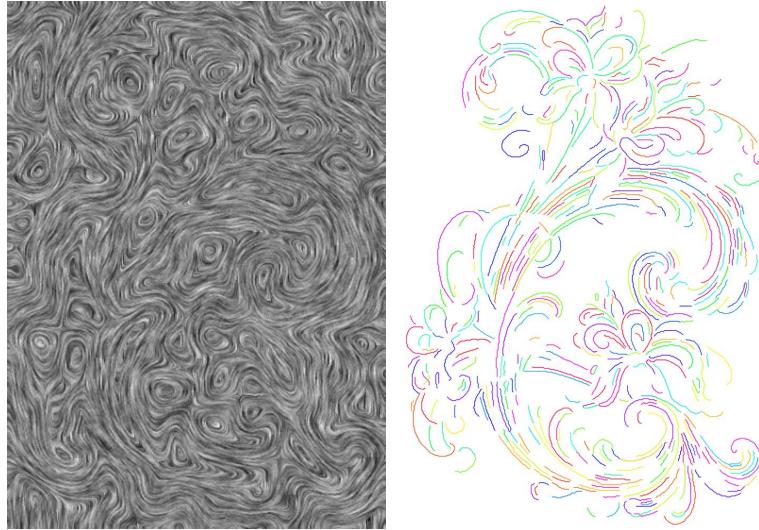


Figure 2.6: Edge Tangent Flow field and coherent lines of a Rosemaling painting. It contains lots of C and S strokes.

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

decomposition of Eq 2.1. Herein, the coherent lines can be computed as follows.

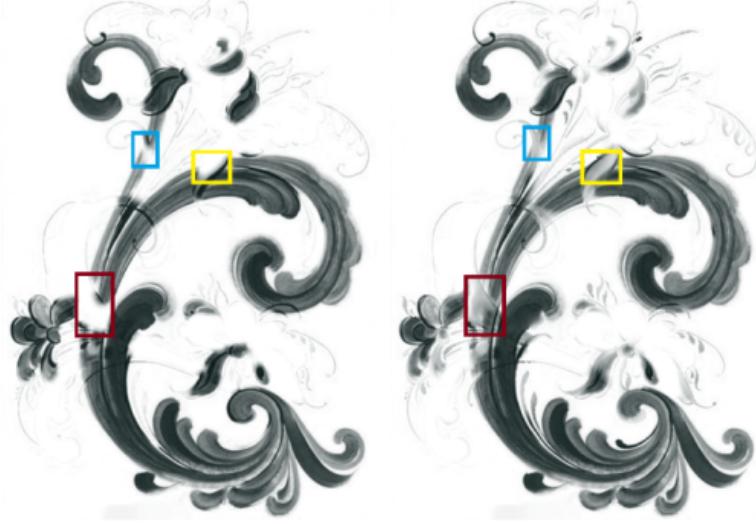


Figure 2.7: Comparison of layer decomposition before and after using E_{edge} in Eq 2.1;

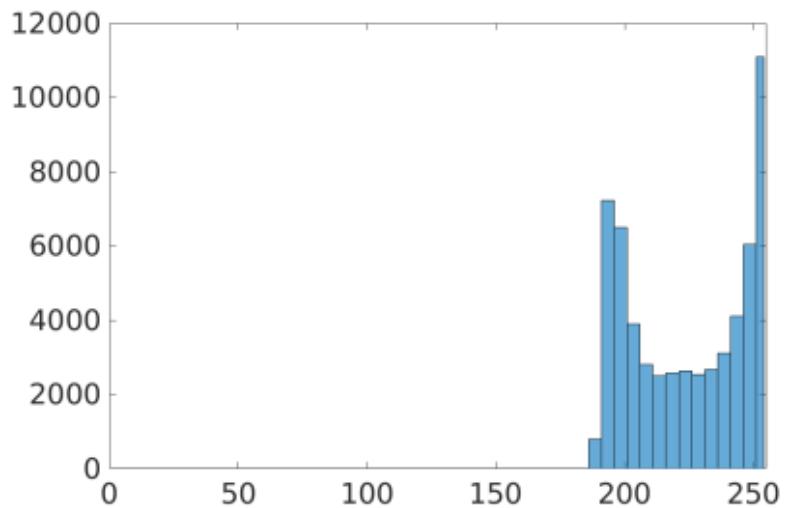
Given a ETF field $t(x)$, the flow-guided anisotropic Difference of Gaussian (DoG) filter is employed, in which the kernel shape is defined by the local flow encoded in ETF field. Note that $t(x)$ represents the local edge direction. It is most likely to make the highest contrast in the perpendicular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



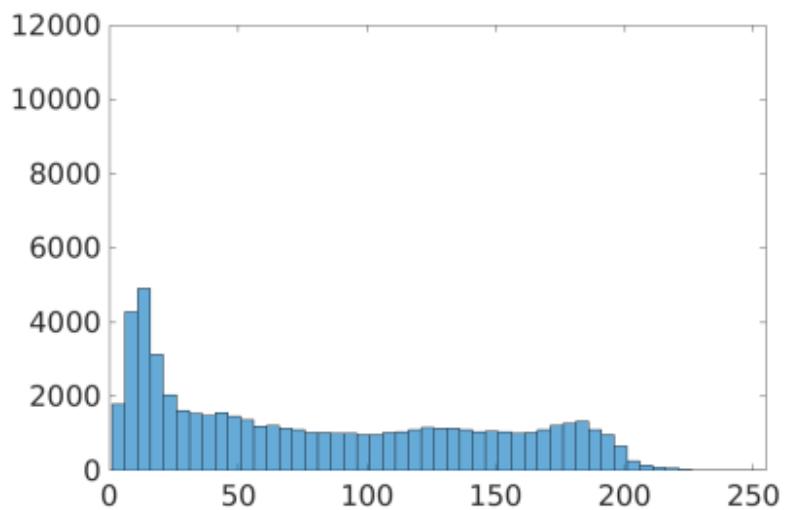
(a) alpha map of layer1

(b) intensity map of layer1

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



(a) histogram of intensity map of layer1



(b) histogram of transparency map of layer1

Figure 2.9: Comparison between transparency and intensity of layer1

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

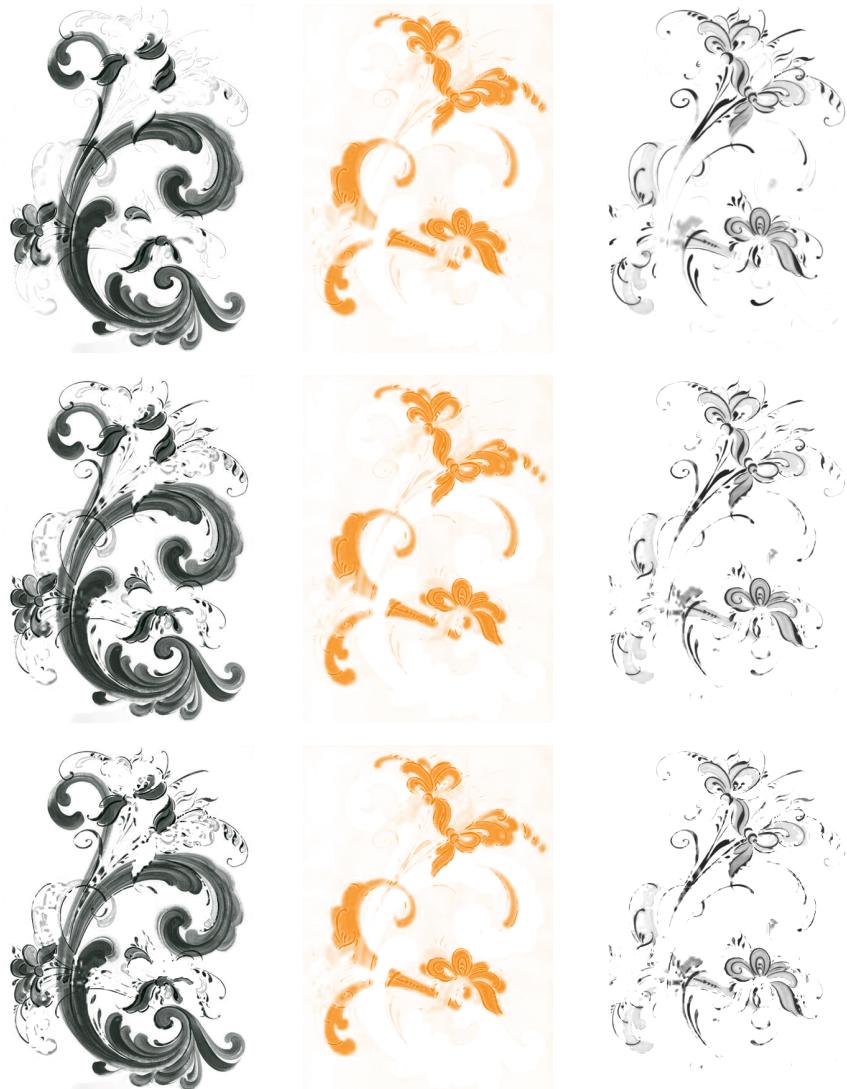
$$E_{edge} = \|LY\|^2 \quad (2.4)$$

where all the opacity α_i are stacked in the vector Y , and L denotes the connection matrix. The eight-connected neighboring rule is utilized to construct the connection matrix L , that is, if two adjacent pixels, i and j , stay on the same coherent line, the item of $L(i, j)$ is set to -1 ; otherwise 0. Figure 2.7a and 2.7b shows that the edges of strokes become visible and complete after involving E_{edge} into Eq 2.1. Accordingly, the layer decomposition of Eq 2.1 is rewritten as,

$$E = \omega_{polynomial}E_{polynomial} + \omega_{opaque}E_{opaque} + \omega_{flow}E_{flow} + \omega_{edge}E_{edge} \quad (2.5)$$

where $\omega_{flow} = 100$, $\omega_{edge} = 20$ for all our examples. Figure 8 shows an evaluation of the effect of changing weights for our one example in which the defaults do not produce the best output.

For comparison, we perform the schemes of Eq 2.1 and Eq 2.5 separately on the same set of brush paintings and compare the root-mean-square-error (RMSE) of the opacity of the coherent lines on each layer shown in Table 1. The RMSE by Eq 2.5 is noticeably less than that by Eq 2.1. 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.



(a) histogram of intensity map of layer1

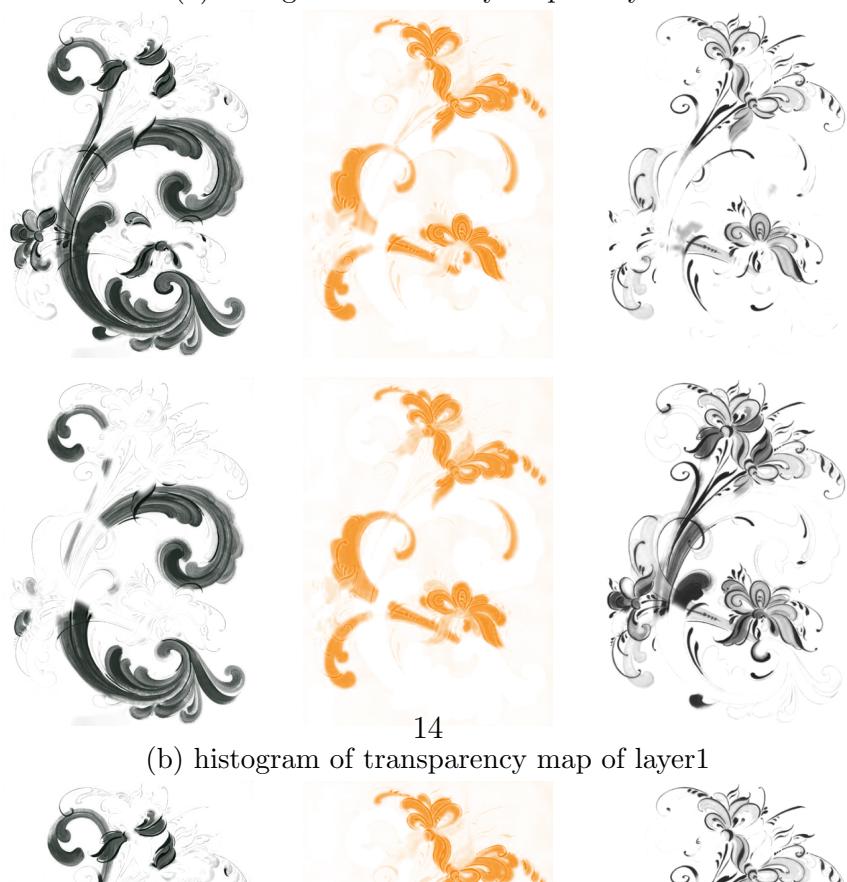


Table 1 Opacity of Coherent Lines on Layers

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.			

Bibliography

- [1] Tim Weyrich, Jia Deng, Connelly Barnes, Szymon Rusinkiewicz, and Adam Finkelstein. Digital bas-relief from 3d scenes. In *ACM Transactions on Graphics (TOG)*, volume 26, page 32. ACM, 2007.
- [2] Jens Kerber, Art Tevs, Alexander Belyaev, Rhaleb Zayer, and Hans-Peter Seidel. Feature sensitive bas relief generation. In *Shape Modeling and Applications, 2009. SMI 2009. IEEE International Conference on*, pages 148–154. IEEE, 2009.
- [3] 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.
- [4] 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.
- [5] Marc Alexa and Wojciech Matusik. Reliefs as images. *ACM Trans. Graph.*, 29(4):60–1, 2010.
- [6] Peter N Belhumeur, David J Kriegman, and Alan L Yuille. The bas-relief ambiguity. *International journal of computer vision*, 35(1):33–44, 1999.
- [7] 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.
- [8] 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.
- [9] 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.
- [10] Thomas Porter and Tom Duff. Compositing digital images. In *ACM Siggraph Computer Graphics*, volume 18, pages 253–259. ACM, 1984.