High Relief from Brush Painting

Yunfei Fu, Hongchuan Yu, Chih-Kuo Yeh, Jianjun Zhang, and Tong-Yee Lee

Abstract—Relief is an art form part way between 3D sculpture and 2D painting. We present a novel approach for generating a texture-mapped high-relief model from a single brush painting. Our aim is to extract the brushstrokes from a painting and generate the individual corresponding relief proxies rather than recovering the exact depth map from the painting, which is a tricky computer vision problem, requiring assumptions that are rarely satisfied. The relief proxies of brushstrokes are then combined together to form a 2.5D high-relief model. To extract brushstrokes from 2D paintings, we apply layer decomposition and stroke segmentation by imposing boundary constraints. The segmented brushstrokes preserve the style of the input painting. By inflation and a displacement map of each brushstroke, the features of brushstrokes are preserved by the resultant high-relief model of the painting. We demonstrate that our approach is able to produce convincing high-reliefs from a variety of paintings(with humans, animals, flowers, etc.). As a secondary application, we show how our brushstroke extraction algorithm could be used for image editing. As a result, our brushstroke extraction algorithm is specifically geared towards paintings with each brushstroke drawn very purposefully, such as Chinese paintings, Rosemailing paintings, etc.

Index Terms—Brush painting, brushstroke, layer decomposition, displacement mapping, high-relief

1 Introduction

As an artistic form, relief spans the continuum between 2D painting and full 3D sculpture, as claimed by many previous works [2], [15], [16], [39], [45]. 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. On this spectrum, 2.5D high relief is closest to full 3D, whereas flatter artworks are described as bas-relief.

Research of relief generation so far has been primarily done based on a 3D model, and meanwhile some methods are based on a single photograph [1], [40], [41], [45] or a line drawing [17], [22], [36], [38] as input. However, how to generate relief from a painting remains unsolved. In particular, we focus on high relief geometric reconstruction from a brush painting, which is useful for supporting applications such as animation, stereoscopy and rotating lenticular posters. Relief generation also has wide applications in making objects such as commemorative medals, souvenirs, and artistic sculptures for the blind. With the commonly and cheaply available 3D printing facilities, there is a growing trend for the need of relief art products. On the other hand, and as input of our relief generation method, the scanned image of paintings are easily accessible on the Internet.

- *C.-K. Yeh, and T.-Y. Lee are with the Department of Computer Science and Information Engineering, National Cheng-Kung University, Tainan City 701, Taiwan, R.O.C. E-mail: simpson.ycg@gmail.com, tonylee@mail.ncku.edu.tw.*
- *Y. Fu, H. Yu, and J. Zhang are with the Media School, Bournemouth University, Fern Barrow, Poole BH12 5BB, UK. E-mail: {yfu, hyu, jzhang}@bournemouth.ac.uk.*

Manuscript received 5 Jan. 2018; revised 13 July 2018; accepted 18 July 2018. Date of publication 25 July 2018; date of current version 31 July 2019. (Corresponding author: Tong-Yee Lee.) Recommended for acceptance by S. Hu. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below. Digital Object Identifier no. 10.1109/TVCG.2018.2860004

Reconstructing a surface from a single 2D image is an ill-posed problem in general. As described in most of the papers on 2D image based reconstruction algorithms, the problem that manifests itself immediately is that there is no complete knowledge of the depth within a single image. Although there have been some image-based relief generation approaches [1], [40], [41], [44], [45], they are not suitable for 2D paintings. These methods prefer to inferring the depth information from a single image rather than others. As a result, the artistic intent is not taken into account. Concerning reproduction or repurpose of an artistic painting, it is crucial that the style of the originals is preserved. In the case of reliefs, although there is no 3D model available, pseudo 3D effect reflecting the style and subtlety is crucial in preserving the artistic essence.

Some research focuses on relief generation from a line drawing [17], [22], [36], [38]. However, maintaining the styles of the brush paintings proves much trickier than simply manipulating the height of the contour lines, since line drawing based methods generate reliefs without consideration for surface details. They are limited to using the information contained in a line drawing, which is not effective for paintings containing information such as color, texture and stroke shape.

A brush painting can be regarded as the union of brushstrokes. The aim of our research is to generate the geometry of a relief from a brush painting. We also argue that because most brush paintings are produced with individual brushstrokes, generating relief surface from each brushstroke would preserve the original features of the painting. Most brush paintings typically contain lots of brushstrokes which exists as individual pieces of art [6], [35]. Differing from the other relief generation methods, our method will preserve this very feature by generating relief surfaces from the brushstrokes individually. This approach nonetheless demands to conquer several challenges. First, unlike photographs, opacity

of brushstrokes is an important feature of a brush painting. Artists are used to varying in color, thickness, texture, consistency and transparency of the paint to achieve certain illusionistic or painterly effects [21]. It is desired to infer the occlusion for relief synthesis, even preserving the pressure of a brushstroke on the relief. Second, unlike the previous 2D image based methods focusing on photographs, a painting does not obey the rules of lighting and shading exactly, which increases the level of difficulty to mimic the details on a relief surface. Third, each brushstroke 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 brushstroke has to be faithfully extracted.

As mentioned above, the previous relief generation methods have demonstrated how to generate reliefs from 3D models, photographs and line drawings. In contrast, our research aims to generate a texture-mapped high relief from an artistic painting. In contrast to the previous 2D image based methods, our algorithm is also the first attempt to generate high reliefs using opacity, and we demonstrate that opacity can preserve more details than intensity. Accompanying this method, we have proposed an algorithm for brushstroke extraction. The major difference between our brushstroke extraction and the previous works is that our algorithm is based on a reformulated layer decomposition algorithm, which makes it capable of extracting overlapped brushstrokes without the prior knowledge of brushstrokes.

2 Related Work

Brushstroke Extraction. To the best of our knowledge, there is a lack of study on extracting brushstrokes from paintings. We give a brief overview of the work related to the relevant topics, i.e., decomposing images into layers and stroke segmentation, which are employed in our implementation. In digital image editing, artists deposit color throughout the image via a set of strokes, which stay at the individual layers. However, scanned paintings and photographs have no such layers. Without layers, simple edits may become very challenging. [32] presents an approach to produce editable vector graphics, in which the selected region is decomposed into a linear or radial gradient and the residual, background pixels. [42] aims at decomposing Chinese paintings into a collection of layered brushstrokes with an assumption that at most two strokes are overlapping and there is minimally varying transparency. [24],[25] present two generalized layer decomposition methods, which allow pixels to have independent layer orders and layers to partially overlap each other. [37] present a layer decomposition method based on RGB-space geometry. They assume that all possible image colors are convex combinations of the paint colors. Computing the convex hull of image colors and per-pixel layer opacities is converted into a convex optimization problem. Thus, their method can work well without prior knowledge of shape and overlap of strokes. Additionally, oil paintings especially van Goghs artefacts, [18] presented the individual supervised/unsupervised extraction schemes to extract brushstrokes for artwork authentication and artist identification purposes. Usually, there is no transparency in the brushstrokes of oil paintings, which is in favor of detecting the boundaries of brushstrokes but it cannot deal with brushstrokes with an unclear boundary or overlapped issues. Currently, most research focuses on stroke segmentation of handwritten characters, such as pen strokes [8]. Pen stroke edges are distinct and can be extracted entirely in handwritten characters. As a result, the basic idea is to describe a stroke by a set of geometric primitives, so that the hand-drawn primitives may be replaced by mathematically precise shapes to produce a neat final result. However, brushstrokes on a painting often contain individual colors and overlap each other. In fact, the strokes may show a low contrast to the others or the background. The edges of strokes are blurred. [42] extracts descriptions of the brushstrokes by using a brushstroke library. However, their approach requires a good amount of prior knowledge of shape and order of strokes. The Most Stable Extremal Regions (MSERs) algorithm [23] was used for establishing correspondence in wide-baseline stereo. [4] introduced the data structure of the component tree in it and further developed it as an efficient segmentation approach, which prunes the component tree and selects only the regions with a stable shape within a range of level sets. The revised version has been widely used in stroke segmentation of handwritten characters [7].

However, it is likely that MSERs may fail in segmentation with the following scenarios: (1) The brushstroke with the intensity very close to the background; (2) two adjacent brushstroke painted by different colors with the similar intensity; (3) overlapped brushstrokes; (4) moreover, like the other existing segmentation approaches, the MSERs algorithm encounters over-segmentation issues as well.

To tackle these challenges, we use opacity of the decomposed layers as input instead of intensity of the image, and the coherent lines method [13] is introduced into MSERs in our algorithm, which both enhances the edges of strokes and preserves the completeness of strokes.

Image Based 3D Modeling: Inferring a 3D shape from a image is a highly under-constrained problem, which requires as much prior knowledge of the 3D shape of an object class as possible, so as to yield reasonable results. When the input is a manmade painting, it becomes more challenging. This is because (1) the man-made paintings may violate the perspective geometry; (2) the input may consist of a lot of strokes that need to be specified by the users; and (3) these strokes are usually inter-related for artistic composition purposes. The closest approach to the topic of brush painting based relief production is the image based method. Since there is a huge volume of previous works in this area, we discuss only the closely related ones below.

Automatic methods. A number of studies have been conducted on automatically reconstructing 3D scene from a single image. [9], [10], [11] extracted the 3D structure of outdoor scenes by segmenting input image and classifying the image into different labels, such as ground, sky, and vertical objects. [33] created the 3D scene using supervised machine learning to model both image depth cues as well as the relationships between different parts of the image. Similarly, [20] estimates the depth of each pixel by semantic segmentation. Shape from shading (SFS) is another related area. Many methods have been developed, such as orthographic SFS with a far light source [19], perspective SFS with a far light source [30], and perspective SFS with a point light source at the optical center [31]. However, these SFS methods assume the image is formed from lighting and

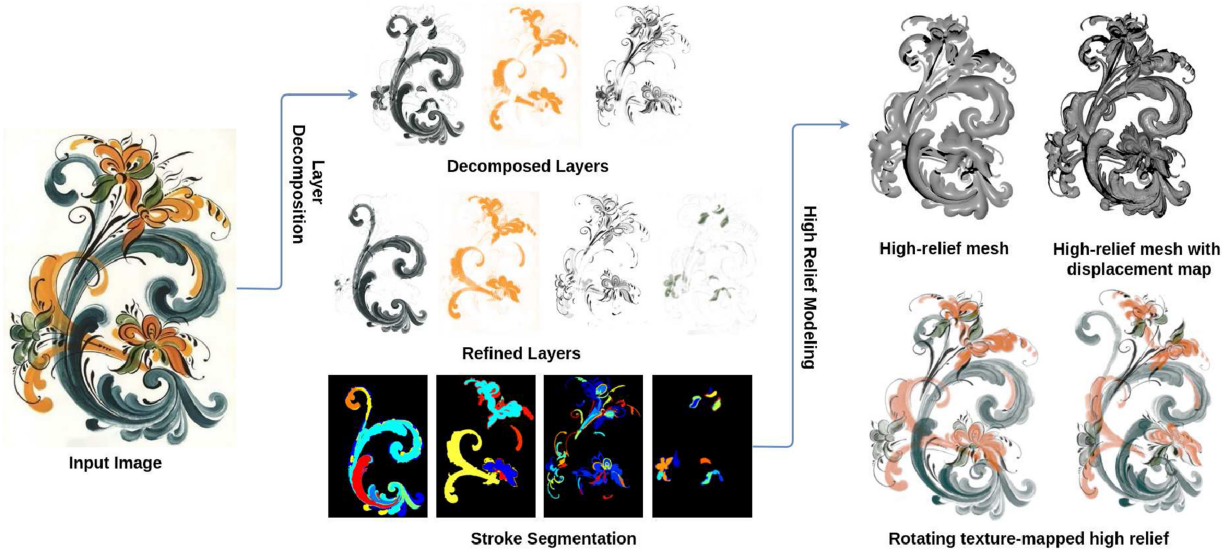


Fig. 1. Overview of the relief generation pipeline.

shading, which may work well for realistic photographs with relatively simple texture rather than paintings. Simply applying the SFS method is not enough to generate relief with acceptable quality for painting images.

In general, fully automatic methods are insufficient to support the reconstruction of complex objects due to insufficient 3D information available in a single image.

User-driven methods. Oh et al.[28] represent a scene as a layered collection of depth images. Two user-assisted tools are employed, based on a painting metaphor, to assign depths and extract layers, and manually inpainted the hidden parts. [46] uses sparse user constraints on input images to generate a smooth shape. Yeh et al.[44] present an interactive user-driven method to reconstruct high-relief 2.5D geometry from a single photo by using user-specified depth cues. The method enables creation of a double-sided organic shape. However, these user-driven methods cannot handle complex shapes and transparency of brushstrokes, and the reconstructed mesh is smooth and so it cannot preserve the brushstroke details.

Sketch based modeling is another research topic relevant to this work. Igarashi et al.[12] presents a sketching interface for designing freeform models by interactive drawing 2D contours. Karpenko and Hughes [14] generate smooth shapes by retrieving suggestive contours from visible-contour sketches. [26] enables users to add, remove, and deform control curves to construct a smooth surface, while the user-drawn strokes stay on the model surface and serve as handles for controlling the geometry. A common drawback of these approaches is that they require tedious specification of control curves to produce the desired shape, and since these methods are generally used for authoring 3D contents, they are not designed for 3D modeling with a fixed reference image.

Line drawing is a drawing made exclusively in solid lines. Rather than generating relief from a photograph of a real scene, some researches focus on relief generation methods from line drawing. Kolomenkin et al.[17] 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 depths, they use smooth interpolation across regions to generate the relief surface. Similarly, line labeling methods have been applied for shape construction from line drawings [36], [38]. 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 relief. [36] proposed a bas-relief generation method consisting 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, our research aims at paintings with brushstrokes, which are different from 2D line drawings. Line drawing based approaches are limited to using information contained in a line drawing, while a brushstroke does not only contain contour lines but also delineates a region, which contain information such as color, texture, opacity. It is crucial to identify and extract brushstrokes from a painting.

3 Overview of Proposed Research

As shown in Fig. 1, a given painting is first decomposed into a set of layers in terms of several specified palette color values (see the decomposed layers in Fig. 1). Second, a new palette color value is determined based on the unclassified regions. Third, the layers are recomputed accordingly in an iterative way (see the refined layers in Fig. 1). To make the regions represent separate brushstrokes, they can be further merged or split. The key point is to extract the overlapped brushstrokes. Overlapped strokes make colors blend. To tackle it, layer decomposition is employed here, which decomposes the painting into a set of translucent layers. In brush paintings, mostly a brushstroke only utilizes a single palette color. Layer decomposition helps classify brushstrokes separately into different layers based on the palette color, so that every layer contains the strokes which are well separated. However, wrong layer decomposition may cut one stroke into two or more layers. It is observed on multiple layers, brushstrokes of such paintings typically follow same patterns. For instance, a

scan of Rosemaling painting employs many C and S brushstrokes, and the color and transparency change very little in the direction of the stroke. However, if assigned the wrong palette color, brushstrokes may appear on multiple layers. We introduce the edge tangent flow (ETF) field and the coherent line [13] to enhance such features in paintings, which are in favor of preserving the completeness of the strokes in every layer and effectively correct the errors due to wrong layer decomposition. Moreover, for the paintings whose strokes can not be clearly decomposed into a limited number of layers, we have developed an iterative scheme to refine the layers. The overlapping regions of multiple strokes with high opacity usually result in the gaps that break the strokes into one or more layers. To tackle this challenge, an inpainting technique is employed here. The coherent line is further involved in the MSERs algorithm [4] again for extracting strokes, which both preserves stroke continuity and removes spurious edges within one layer.

Furthermore, to generate the displacement maps of the strokes individually, we perform shape from shading(SFS) on the opacity of the paintings instead of the intensity, since the opacity has a bigger range than the intensity. The 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 transparency of brushstrokes, to the surface of the brushstroke models. Using the inflation method proposed in [44], we inflate the surface for each brushstroke and combine it with the displacement maps. The shapes of all the strokes are then arranged to form the desired high-relief. In general, the background plane should be unchanged. Thus, generating strokes individually not only benefit the composition of reliefs, but also avoids this technique issue.

4 Layer Decomposition

Digital painting with different layers is an integral feature of digital image editing software, such as Photoshop and Sketchbook. Layers offer an intuitive way to edit the color and geometry of components and localize changes to the desired portion of the final image. Without layers, brushstroke segmentation becomes extremely challenging, since they can overlap and blend with each other. In general, each layer represents one coat of a painting with a 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 brushstrokes in layer decomposition. To this end, we modify the layer decomposition algorithm in [37] by involving the coherent lines [13] in our implementation. In the following we first address the layer decomposition algorithm briefly and then discuss our modification.

4.1 Layer Decomposition Scheme

The A over B compositing and blend mode in [29] described that when the pixel A with color and translucency is placed over the pixel B with color and translucency , the observed color is,

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

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 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 ith layers color, α_i is the opacity of C_i , the background color C_0 is opaque, and 3 or 4 depending on the number of channels (RGB or RGB-α). 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 is the spatial gradient of opacity in the ith 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 discontinuity at the edges.

The users may specify the layer order in advance, as well as the number of layers, n. The opacity for every layer may be solved by minimizing the following combined cost function,

$$E = \omega_{polynomial}E_{polynomial} + \omega_{opaque}E_{opaque} + \omega_{spatial}E_{spatial}, \tag{2}$$

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

4.2 Modified Layer Decomposition

To enhance the smoothness and completeness of edges, the coherent line drawing technique in [13] is introduced to Eq. (2) , which is a flow-guided anisotropic filtering framework. Fig. 2 shows the edge tangent flow (ETF) field of a Rosemaling painting. First, we involve the ETF field into $E_{spatial}$. The ETF field is defined as,

$$t^{new(x)} = \frac{1}{k}\sum_{y\in\Omega(x)} \varphi(x,y)t^{current}(y)\omega_s(x,y)\omega_m(x,y)\omega_d(x,y). \tag{3}$$

where $t(x)$ 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

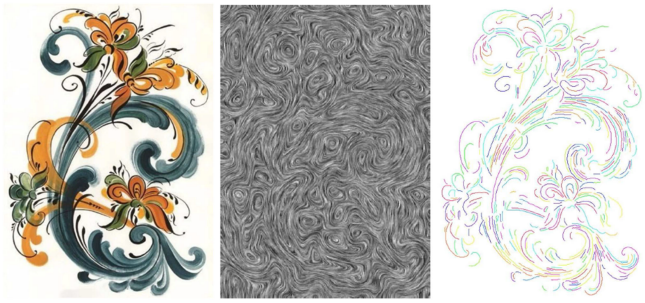


Fig. 2. Edge Tangent Flow field and coherent lines of a Rosemaling painting. It contains lots of C and S strokes.

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., while suppressing swirling flows. In addition, the sign function is employed to prevent the swirling artefact as well.

Involving ETF filed of Eq. (2) in $E_{spatial}$, the smoothness penalty is rewritten as,

$$E_{flow} = \frac{1}{n}\sum_{i=1}^{n} \|t^{new}\|(\bigtriangledown_{\theta}\alpha_i)^2, \tag{4}$$

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. (2) instead of $E_{spatial}$, the brushstrokes become complete and smooth, which can be noted in Fig. 3.

Second, the coherent lines as the constraint of brushstroke edges are involved in layer decomposition of Eq. (2). Herein, the coherent lines can be computed as follows. 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 brushstrokes.

To preserve the brushstroke 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,

$$E_{edge} = \|LY\|^2, \tag{5}$$

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. Fig. 3d shows that the brushstrokes become visible and complete after involving E_{edge} into Eq. (2). Accordingly,

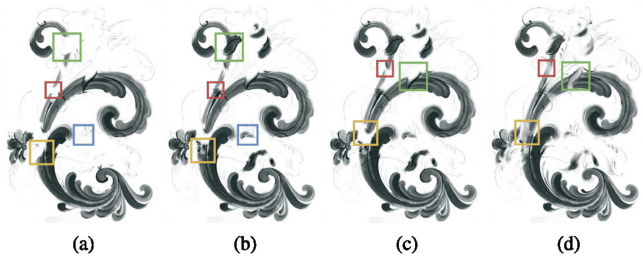


Fig. 3. Comparison of layer decomposition by Eqs. (2) and (6) at the 2nd layers. (a) and (b) show the results by using $E_{spatial}$ and E_{flow} in Eq. (2); (c) and (d) show the results before and after using E_{edge} in Eq. (6).

the layer decomposition of Eq. (2) is rewritten as,

$$E = \omega_{polynomial}E_{polynomial} + \omega_{opaque}E_{opaque} + \omega_{flow}E_{flow} + \omega_{edge}E_{edge}, \tag{6}$$

where $\omega_{flow} = 10, \omega_{edge} = 1$ for all our examples. For comparison, we perform the schemes of Eqs. (2) and (6) 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 is defined as, $RMSE = \sum_{i=1}^{N}\sqrt{\sum_{j=1}^{n_i}(\alpha_{i,j} - \bar{\alpha_i})/n_i}$. N donates the number of coherent lines, where the $\bar{\alpha_i}$ is the average opacity of n_i pixels on the ith coherent line, and $\alpha_{i,j}$ is the opacity value of jth pixel on the ith coherent line. The RMSE by Eq. (6) is noticeably less than that by Eq. (2). 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. Moreover, the resulting layer by Eq. (6) is shown in the upper row of Fig. 4.

4.3 Iterative Scheme

The decomposed layers can be separated into the background and foreground(brushstroke regions) by thresholding opacity. It can be noted in the first row of Fig. 4, there are some regions shared in multiple layers since such shared regions have visible opacity at the multiple layers. As there are a lack of layers, the colors of the shared regions have to be yielded by blending the colors of the current multiple layers. Moreover, it can be noted that the shared regions may be categorized into two kinds, one is the region overlapped by multiple brushstrokes with palette color, and the others are the isolated ones, as shown in Fig. 5. The former are always merged into the other regions within some layers, while the latter are always isolated and have larger opacity than threshold in all the layers. It is natural to view the isolated regions as potential strokes. We therefore set the average color of the largest isolated region as a new palette color, and then re-compute the opacityof each layer by solving Eq. 6. Fig. 4 shows the comparison of before and after adding a new layer. Furthermore, to remove the isolated regions, it can be achieved by adding the new layers in an iterative way. Usually, after 1 or 2 iterations, there is no isolated region to appear in the new layer. This can be noted in Fig. 5, that is, after 1 iteration the isolated regions have a distinct change.

4.4 Brushstroke Completion

As shown in Fig. 4, some brushstrokes may have other opaque brushstrokes overlapped above, which bring about the gaps to break the brushstrokes within a layer. Obviously, to

TABLE 1
Opacity RMSE of Coherent Lines on Layers

Painting ID	Number of Layers	Layer	Opacity RMSE of coherent lines By Eq. 2	By Eq. 6
Rooster (Fig. 21, row 1)	4	1	25.12	15.39
		2	8.89	5.92
		3	15.74	9.63
Man (Fig. 21, row 2)	4	1	38.41	26.33
		2	21.28	16.37
		3	5.26	4.19
Bird (Fig. 21, row 3)	4	1	8.24	6.12
		2	7.15	4.25
		3	9.24	6.71
Lotus (Fig. 21, row 4)	4	1	16.44	10.25
		2	20.54	15.04
		3	10.56	5.87
Lotus2 (Fig. 22, row 3)	4	1	17.58	13.87
		2	19.78	8.72
		3	22.47	17.24
Rosemaling1 (Fig. 16, row 1)	5	1	27.11	21.42
		2	17.52	14.58
		3	16.2	17.99
		4	19.24	11.51
Rosemaling2 (Fig. 16, row 2)	5	1	15.44	12.84
		2	21.22	24.71
		3	25.72	19.95
		4	24.98	21.36
Rosemaling3 (Fig. 16, row 3)	7	1	16.53	11.94
		2	15.71	12.15
		3	21.21	16.19
		4	19.48	15.61
		5	15.21	9.21
		6	11.01	4.85
Van gogh1 (Fig. 14 , row 1)	5	1	10.18	28.11
		2	12.27	14.11
		3	22.34	13.41
		4	11.85	8.65
Van gogh2 (Fig. 22, row 2)	4	1	15.89	14.25
		2	19.54	15.49
		3	17.21	12.98

make brushstrokes complete and smooth, these gaps need to be filled. It is natural to involve user interventions, such as manual masks or sketches, which specify the gaps to be filled.

Algorithm 1. Iterative Scheme

Input: N decomposed layers
Output: $N+1$ decomposed layers
1 Binarize each decomposed layers;
2 Detect the isolated regions from the binarized layers;
3 Select average color of the largest isolated region as new palette color C_{n+1} ;
4 Re-compute the decomposition by solving Eq. (6) with C_{n+1};

Once the overlapped regions are determined, we employ the patch-based inpainting techniques [43] here, that is, within one layer, the gap is specified by a mask, while the patches are extracted from the outside of the gap in all layers and are utilized to create the patch dictionary. Then, the gap is filled through iteratively projecting each patch of the layer to its nearest neighbor in the dictionary. Fig. 6 shows that such patch-based inpainting methods can effectively deal with the scenario of a big gap. For fair demonstration and comparison, other than the inpainting shown in Fig. 6, we do not involve the inpainting technique for the other examples.

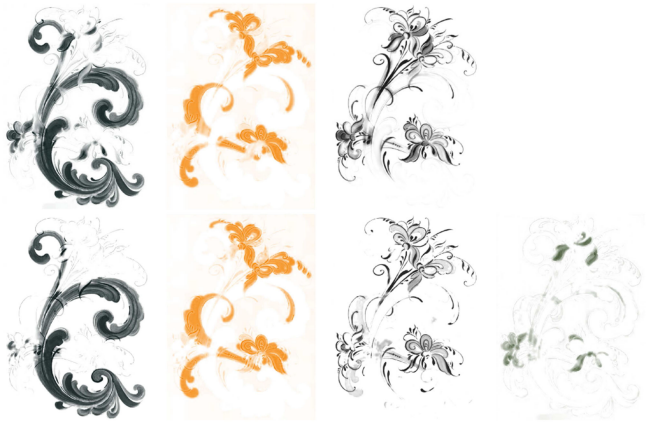


Fig. 4. Comparison of before and after adding a new layer. The upper row shows the decomposed layers, while the below row shows those layers after adding a new layer. Each column shows the decomposed layers with a corresponding palette color. The last column of the below row shows the new layer.

Fig. 5. The masks of the shared and isolated regions when there are 4 and 5 layers respectively. (a) Shared regions (4 layers). (b) Shared regions (5 layers). (c)Isolated regions (4 layers). (d)Isolated regions (5 layers).

5 Extraction of Brush Strokes

Brushstrokes normally follow certain rules in a brush painting and they vary depending on the style of the painting. Here we discuss how we make use of such rules. Rosemaling paintings

Fig. 6. Illustration of inpainting on a layer.

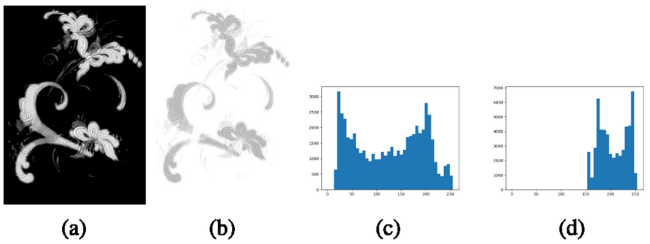


Fig. 7. Comparison of the intensity (a) and opacity (b) of a layer with histograms(c)(d).

usually use subtle and vibrant colors to enhance color contrast between overlapped strokes. As a result, the overlapped strokes tend to be classified into different layers. Extracting brushstrokes within one layer is easier than directly from the input painting. We employ the MSERs proposed in [4] and [27] to extract brushstrokes since it is invariant to affine intensity changes. MSERs algorithm requires a distinct difference between background and foreground while allowing a small variation of intensity within the selected stroke region. Usually, the strokes on the decomposed layers satisfy this requirement.

However, it is likely that MSERs may fail in segmentation with the following scenarios, (1) two adjacent regions with the similar intensity; (2) the region with a high transparency. Moreover, like the other existing segmentation approaches, the MSERs algorithm encounters an over-segmentation issue as well. To tackle these challenges, the coherent lines [13] are introduced into MSERs, which both enhances the edges of strokes and preserves the completeness of strokes. For completeness sake, we briefly address the MSERs algorithm and then address our modification.

5.1 MSERs Algorithm

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

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

where ∂S denotes the boundary of S. The extremal regions can be detected by changing threshold. With given intensity threshold g, all pixel with intensity larger than g is black, otherwise is set to white. By changing threshold g, these black and white regions may further split or merge indicates the set of all extremal regions. 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 ith 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.

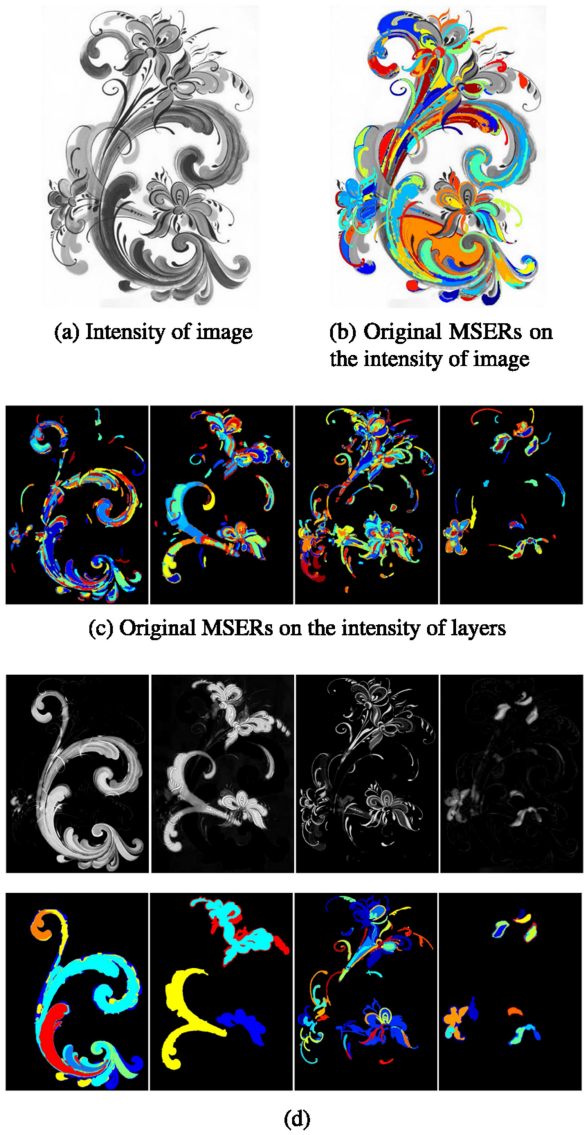


Fig. 8. Comparison of segmentation results by the original MSERs and modified version. (a) Intensity of input. (b) The original MSERs on the intensity of image. (c) The original MSERs on the intensity of layers. (d) Opacity of layers, and MSER regions on four layers by the modified MSERs.

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, MSERs may fail in segmentation with the following scenarios, (1) the region with the intensity most close to the background; (2) two adjacent regions with the similar intensity; (3) the overlapped brushstrokes. The opacity of the image, α, is always independent of the intensity of the image. It is likely to avoid the abovementioned scenarios if segmentation can be performed on α. We perform the layer decomposition of Eq. (6) on a brush painting and show the intensity and opacity of one layer associated with the individual histograms in Fig. 7. It can be noted that the opacity of the layer contains richer layered details than the intensity. Moreover, Fig. 8 also shows that the opacity of the layers is more suitable for

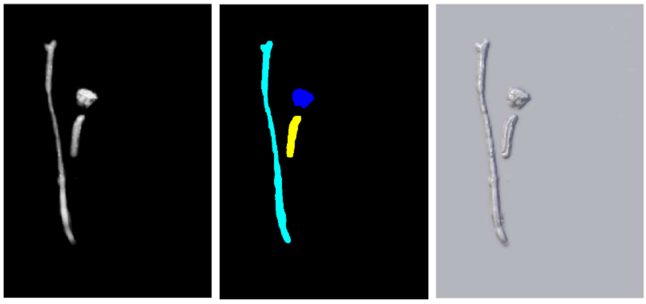


Fig. 9. Displacement map generation from brushstrokes. (a) Opacity of one layer. (b) Extracted three brushstrokes. (c) Generated displacement map from three brushstrokes.

stroke segmentation than the intensity. The first modification is to perform MSERs on the opacity of every layer.

We aim at the scenarios of two adjacent regions with the similar opacity. When the extremal region is growing up through changing threshold, it is feasible to restrict the region by introducing the coherent lines. According to the definition of the extremal region, the boundary of region S should satisfy,

$$\forall p \in S, \forall z \in \bar{S}, \forall q \in \partial S \longrightarrow I(p) \geq I(q) \text{ and } I(q) \leq I(z), \quad (7)$$

where $\bar{S}$ denotes the complement of S. The second modification is to simply modify the opacity of layers, that is, overlapping the coherent lines with the layer and then changing the opacity of coherent lines to the smallest value in the layer. To deal with the over-segmentation issue, the coherent lines play an important role. Given a region S, we modify the area change rate γ as,

$$\gamma(S_i^g) = \frac{S_j^{g-\Delta} - S_k^{g+\Delta}}{S_i^g} + \frac{Q_j^{g-\Delta} - Q_k^{g+\Delta}}{Q_i^g} - \left(1 - \frac{Q_i}{S_i^g}\right), \quad (8)$$

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.

Fig. 8 shows the segmentation results by the modified MSERs, which correspond to brushstrokes. It can be noted that performing MSERs on the intensity of image or the intensity of decomposed layers inevitable yields over-segmentations. Performing the modified MSERs on the opacity of layers, the strokes tend to be 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 the background.

6 Relief Generation

In brush paintings, each brushstroke is often introduced to depict something specific in the real world [42] . Thus, the output of our stroke-based decomposition of these paintings is a set of graphical objects that are meaningful with regard to the set of real objects the paintings depict. It is natural to generate depth information from brushstrokes, here, we demonstrate how to generate relief from brushstrokes.

In our implementation, we employ the orthogonal SFS [30] algorithm on the segmented brushstrokes. The brightness equation used in the SFS algorithm is expressed as,

$$I(x) = \frac{1}{\sqrt{1 + |\bigtriangledown h|^2}}. \quad (9)$$

$I(x)$ 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 brushstrokes are usually 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 brushstrokes is independent of the color (see Fig. 7). Each stroke has an appropriate distribution of opacity, which is in favor of a layered look. On the other hand, the intensity of stroke is determined by several facts: the color and opacity of brush stroke, background color and the overlapped strokes if there is any. Since colors of each layer are determined, based on Eq. (1), intensity of brush stroke is a linear function of opacity. Displacement map generated from intensity would be unavoidably influenced by the background color, and the feature of the overlapped strokes, which is unwanted in each brushstroke mesh. Generating displacement map from opacity would better preserve the feature of brushstrokes, so we reformulate the equation :

$$\alpha(x) = \frac{1}{\sqrt{1 + |\bigtriangledown h|^2}}, \quad (10)$$

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

$$|\bigtriangledown h| = \sqrt{\frac{1}{|\alpha(x)|^2} - 1 + \Delta}, \quad (11)$$

where Δ is a positive displacement which set to 0.2 by default. This modification may make the surface inflated. By using fast marching algorithm [34] to solve this equation, we can generate a displacement map for each brushstroke. Fig. 9 shows that we input the opacity map of a decomposed layer and extract three brushstrokes from it, which are depicted in different color. Then, applying Eq. (11) to these three brushstrokes generates the individual displacement maps. By merging them together, the displacement map of the entire input painting is generated accordingly. Moreover, we apply the inflation method proposed in [44], i.e., controlling the makeup cues on extracted brushstroke regions to generate inflated smooth surface for each brushstroke. As shown in Fig. 10, two very simple user-markup cues are used to guide the inflation of burshstroke regions. The slope cues(the green icon) enable brushstroke regions to pop up from the image plane, and by dragging the arrow of the cues, users can control the slope magnitude of the regions. A curvature cue(red icon) constrains the local mean curvature of a extracted brushstroke which allows users to manipulate the inflation of the local shape. Similarly, the curvature amount is corresponding to the length of the curvature cues. Up to now, we reach the desired high relief model. Fig. 10 shows that the extracted brushstrokes are mapped to relief surfaces respectively.

Fig. 10. (a) Inflation with markup cues (b) Smooth high relief surface. (c) High relief with displacement map. (d) Texture-mapped high relief (rotated 30 to 45 degree from the original viewing direction).

7 Results and Analysis

We used the published codes of layer decomposition and MSERs, which are available on GitHub (at: https://github.com/CraGL/Decompose-Single-Image-Into-Layers; and https://github.com/idiap/mser), and performed the proposed approach on various brush paintings, including Rosemaling, van Gogh oil painting and Chinese brush paintings. All the tests were performed on a 6-core of 3.33 GHz Intel Core Xeon CPU with memory of 32 GB(RAM).

In our implementation, the parameters in Layer decomposition, ETF field, and MSERs are set the default values as in the original codes. Table 3 further shows the running time of the proposed approach. Compared to the performance in [37] and [27], there is no distinct difference. To demonstrate the extracted strokes, we did three further tests, including recoloring paintings, inserting objects and animating strokes. Our implementation is not multithreaded. All the resulting images/videos are available in the supplementary materials, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TVCG.2018.2860004.

In our decomposition, the number of layers (color palette size) is chosen based on users' observation. Each layer aims to represent one coat of a painting with a single palette color and we wish to have a manageable number of layers. Too small palette size cannot generate all color in the input image, and could result in wrong brushstroke segmentation (see Fig. 11a). Too large palette size would result in an unwieldy number of layers. To make our method more accessible, same as [37], our decomposition (Eq. (6)) is computed in a multi-resolution manner: the initial solution is computed based on recursively downsampled input images and we upsample the solution as initial guess for larger image (finally, the input image size). By such manner, user can quickly preview the low resolution decomposed results (seconds for a 100x100 pixels image) which helps users to experiment different layer numbers more efficiently. In other words, users can quickly choose a layer number based on the preview. From the same input image (2nd row of Fig. 16a), we show the effect of changing the layer number (see Fig. 11).

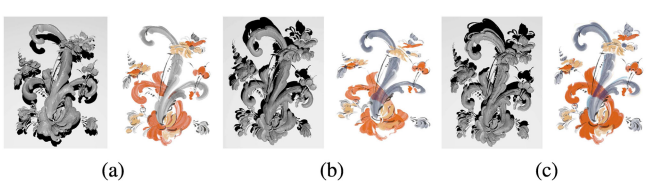


Fig. 11. The effect of changing the layer number. (a) High relief generated from 4 layers. (b) High relief generated from 6 layers. (c) High relief generated from 9 layers.

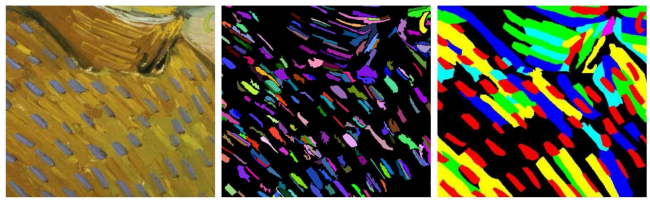


Fig. 12. Brush strokes extraction in Li et al.'s work [18]. (a) Input painting region. (b) Brushstroke extracted by Li et al.'s method [18]. (c) Manually marked brushstrokes.

We have also compared our method against several existing alternatives, which we obtained or reproduced their implementations. Specially, we experimented with the stroke automatic extraction of Li et al.[18], and high relief generation approach of Yeh et al.[44].

7.1 Comparison with Li at al.'s Method

To the best of our knowledge, there is lack of study on automatic brushstrokes extraction. Li et al.[18] presented a automatic brushstroke extraction method based on seed growing technique for Van Goghs paintings. To numerically evaluate their brushstroke extraction algorithm, they manually marked brushstrokes using 10 sample regions from van Gogh's paintings, see Fig. 12. Similarly, we also manually marked brushstrokes using 10 sample regions from the same collection of [18]. On average, there are 120 manually marked brushstrokes per painting, and brushstrokes are marked by artists with sufficient experience.

In terms of the marked samples, they defined two parameters, i.e., valid rate and detection rate, which are utilized to evaluate the accuracy of the extracted brushstrokes, and therefore are applied to the comparison between their method and ours.

Valid rate: the percentage of valid automatically extracted brushstrokes.

Detection rate: The percentage of detected manual brushstrokes. Moreover, for a fair comparison, we tested the van Gogh paintings given in their work and show the results in Figs. 13 and 14.

As shown in Table 2, the valid rate and detection rate of our method is noticeably higher than Li et al.'s method [18] and default MSERs algorithm. The painting ID (F-numbers)

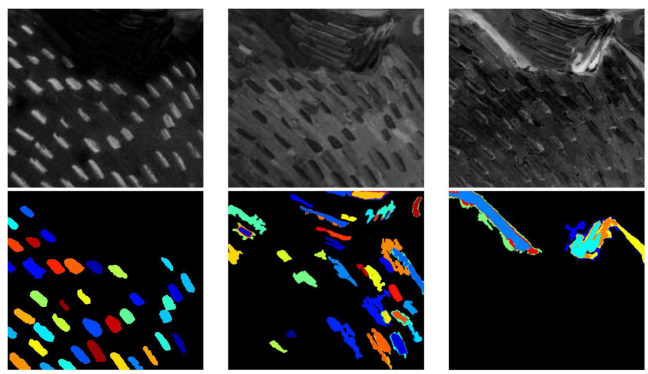


Fig. 13. Brush strokes extraction by our method; the upper row shows the opacity maps of each layer.

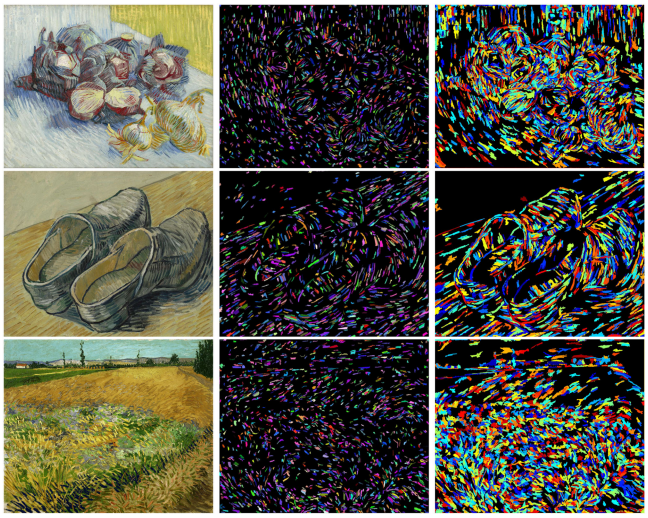


Fig. 14. (a) Input image. (b) Brushstrokes segmented by method in [18] (c) Brushstrokes extracted by our method on three decomposed layers.

of van Gogh's paintings is based on the catalogue numbers in [5]. Additionally, compared to Li et al.'s method [18], our method is based on layer decomposition, it has better performance in the following scenarios: (1) the brushstroke with the intensity very close to the background; (2) two adjacent brushstroke painted by different colors with the similar intensity; (3) overlapped brushstrokes.

7.2 Comparison with Yeh et al.'s Method

We compare our method with the most closely-related work by Yeh et al. [44], mainly on four aspects: segmentation, local layering, inflation, and texture-map. The method proposed in [44] was designed for high-relief 3D models from a single input image of organic objects with nontrivial shape profile. Our high relief generation process is similar to [44]'s method. The distinct difference is that the interactive user-driven region segmentation is replaced with the brushstroke extraction of each layer. After that, combining the SFS with the inflation, we re-rendered the high-relief based on the individual extracted brushstrokes instead of the raw image.

*Segmentation.*Yeh et al.[44] extended the standard pixel-based graph-cut method [3] to be (2D) mesh-based to produce sharper segmentation boundary, in which the user scribbles

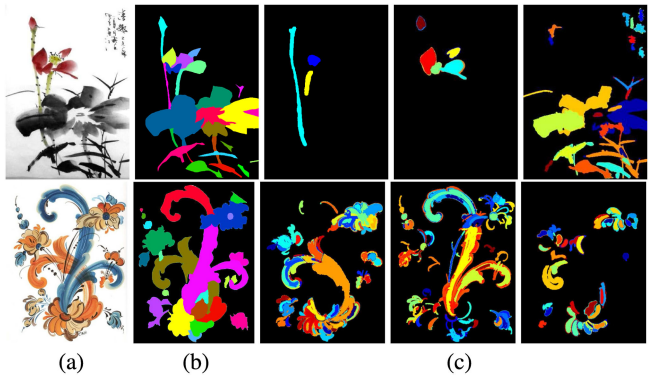


Fig. 15. (a) Input image. (b) Brushstrokes segmented by method in [44]. (c) The last three columns show brushstrokes extracted by our method on each decomposed layers.

foreground and background labels on the input image to subdivide it into different regions. However one painting may contains hundreds of brushstroke. To successfully segment the brushstrokes, the users must specify a tedious amount of scribbles. It is likely to fail to segment thin and fine structures commonly seen in brushstrokes. As a comparison, our method is automatic and able to preserve the thin and complex shapes of brushstrokes, as showing in Fig. 15.

Inflation. [44]'s method is designed for reconstructing smooth and organic shapes, assuming the inflation model to be smooth. To preserve the fine details of the brushstrokes, we apply opacity to generating the surface, and the displacement map to the high relief, which better maintains the details of brushstrokes. Fig. 16 demonstrates the difference of the relief surfaces by our method and [44]'s.

The overall impression of [44]'s results is fine, but as a comparison, our approach can better preserve the thin strokes and the fine details of brushstrokes.

Local Layering. Each brushstroke covers a region on the canvas and they may overlap with each other, some quite heavily in a painting. In order to achieve faithfully complete brushstrokes, [44] requires clear edge of region to form the T-junctions. For brushstrokes with a complex shape, users have to manually divide the edge and label the layers. In our method, the overlapped brushstrokes with different colors can be extracted by layer decomposition, in which the information of each brushstrokes is retained. Fig. 17 clearly shows that the overlapped brushstrokes can be extracted by our method.

TABLE 2
Brushstrokes Extraction Evaluation and Comparison

	Valid Rate (%)			Detection Rate (%)		
Painting ID	[18]'s method	Default MSERs	Our method	[18]'s method	Default MSERs	Our method
F218 (Glass with yellow roses)	42.7	79.8	88.1	21.6	33.7	48.5
F248a (Vase with Gladioli)	75.4	79.4	90.2	78.8	82.1	88.1
F297 (Skull)	57.9	67.2	75.3	52.0	61.0	75.1
F374 (Red Cabbages and Onions)	58.2	67.9	75.9	63.2	60.8	82.4
F386 (Still live with potatoes in yellow bowl)	73.7	64.1	82.4	68.4	59.2	90.2
F415 (Seascape near Les Saintes-Maries-de-la-Mer)	46.9	50.3	58.1	60.0	80.1	92.1
F518 (The little Arlesienne)	60.7	55.7	71.2	75.2	86.1	78.9
F538 (Portrait of Camile Roulin)	49.0	58.4	84.2	44.9	61.4	81.3
F572 (Willous at Sunset)	83.9	68.9	82.4	65.6	71.2	87.0
F652 (Pine with female figure in Sunset)	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

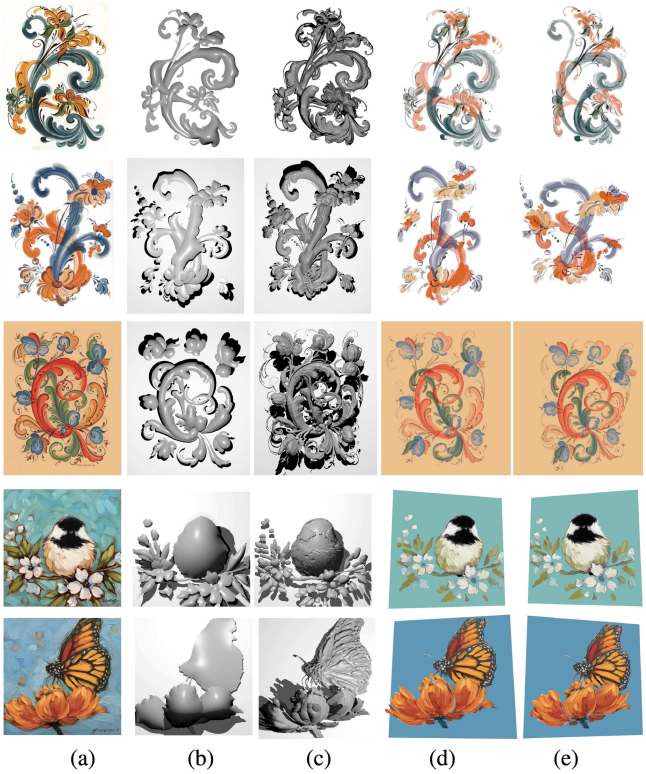


Fig. 16. (a) Input image. (b) High reliefs from [44]'s method. (c) High reliefs generated by our method. (d,e) Texture-mapped high relief by our method.

Texture-Map. Compared to [44], our method exploits the opacity of brushstrokes to transfer the details to high relief, which effectively preserves the artistic feelings of the paintings. Fig. 18 shows the texture-mapped high-relief results that are generated from a painting. Herein, Fig. 18a shows the input (single) image, while Figs. 18b and 18c show views of the models with large rotation from the original view. As shown in Fig. 18b, [44] simply uses the raw image as texture for each inflated brushstroke region, which is inappropriate, especially for the overlapped brushstroke region. Instead of using the raw image as texture, the extracted brushstrokes with opacity values are mapped to the relief surface in our method, which can better show the features and inter-relations of brushstrokes on high relief. More results are shown in Fig. 21.

Fig. 17. Extraction of overlapped brushstrokes. (a) Input images. (b) The segmentation by Yeh et al.[44]'s method (blue rectangles indicate the overlapped regions of brushstrokes). (c) The extracted brushstrokes on decomposed layers by our method.

Fig. 18. (a)Input image. (b) Texture-mapped high relief by method in [44]. (c) Texture-mapped high relief by our method.

Running Time. More specifically, we compare the running time in segmentation and relief generation between our method and [44]'s method. For segmentation, our method is less time-consuming, especially for a input painting containing many brushstrokes. Relief generation in [44]'s method includes several steps: layering, completion, user annotation and stitching, which requires more time for inflation on a single region than our method. As shown in Table 3, our method is less time-consuming, even though we have to inflate more regions than [44].

Limitations. In our implementation, we employ layer decomposition in order to split the overlapped brushstrokes into different layers, so that the strokes can be easily segmented within one layer. However, the main challenge is

TABLE 3
Performance

	[44]'s method				Our Method					
	Segmentation				Segmentation			Relief Generation		
Painting ID	Stroke number	Time	Relief generation	Total time	Stroke number	Runtime of Layer	Runtime of MSERs	Runtime of SFS	User annota- tion& Inflation	Total time
Rooster (Fig. 21, row 1)	112	35 m 31 s	9 m 44 s	45 m 15 s	292	311.25 s	0.64 s	8.68 s	11 m 37 s	16 m 57 s
Man (Fig. 21, row 2)	53	18 m 12 s	8 m 12 s	26 m 24 s	110	162.21 s	0.85 s	7.58 s	10 m 43 s	13 m 34 s
Bird (Fig. 21, row 3)	28	08 m 21 s	4 m 25 s	12 m 46 s	38	70.54 s	0.32 s	6.54 s	5 m 12 s	6 m 39 s
Lotus (Fig. 21, row 4)	34	15 m 01 s	7 m 21 s	22 m 22 s	45	112.21 s	0.34 s	8.25 s	8 m 31 s	10 m 32 s
Lotus2 (Fig. 22, row 3)	120	39 m 08 s	10 m 36 s	49 m 44 s	258	387.11 s	0.85 s	17.21 s	8 m 21 s	15 m 1 s
Rosem aling1 (Fig. 16, row 1)	42	20 m 22 s	8 m 09 s	28 m 31 s	114	164.87 s	0.75 s	7.26 s	9 m 19 s	12 m 12 s
Rosemaling2 (Fig. 16, row 2)	62	25 m 27 s	17 m 54 s	43 m 21 s	220	157.32 s	0.45 s	5.85 s	15 m 5 s	17 m 49 s
Rosemaling3 (Fig. 16, row 3)	35	19 m 45 s	6 m 15 s	26 m 0 s	105	60.23 s	0.55 s	5.92 s	7 m 04 s	8 m 11 s
Bird2 (Fig. 16, row 4)	42	18 m 35 s	5 m 31 s	24 m 06 s	85	110.48 s	0.82 s	7.11 s	12 m 13 s	14 m 11 s
Butterfly (Fig. 16, row 5)	6	5 m 54 s	3 m 15 s	9 m 09 s	30	50.10 s	0.44 s	5.86 s	6 m 24 s	7 m 21 s

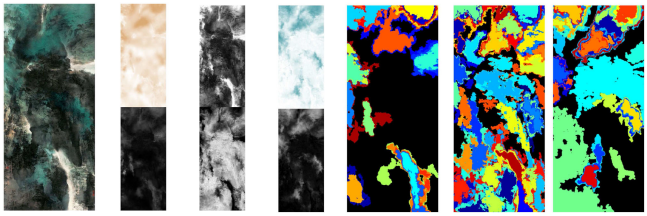
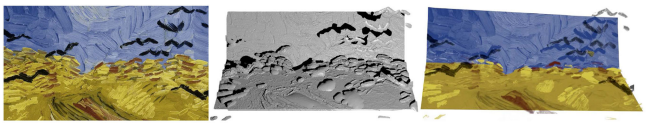


Fig. 19. Brush strokes highly blended with painting colors.



(a) (b) (c)

Fig. 20. (a) A region from a van Gogh's painting (F779, Wheatfield with Crows). (b) High relief model. (c) Texture-mapped high relief.

that when the overlapped strokes share the same color in some brush paintings, they would appear on the same layer and might be regarded as one brushstroke. Rosemaling painting designs use subtle and vibrant colors to enhance the color contrast between adjacent brushstrokes. Moreover, for a layered look, transparency is applied and is variable with a big scope. Thus, layer decomposition benefits segmenting the overlapped brushstrokes on Rosemaling paintings. As opposed to it, some brush paintings do not emphasize the use of vibrant colors, even the adjacent strokes may share the same color in some paintings. The boundary of the overlapped brushstrokes may be very blurred. This may result in stroke segmentation failure. Even some paintings can be too complex for proper segmentation, e.g., brushstrokes highly blended with painting colors; We cannot extract the meaningful regions for relief generation as shown in Fig. 19. Our method also assumes the brushstroke has generally uniform color, which may fail to extract brushstrokes which delicate textures with distinct different colors.

Our segmentation does not involve semantic information, while a van Gogh's painting may employ hundreds of strokes to represent a semantic region, such as a wheat field (Fig. 22b), a onion (first row in Fig. 14a). In our method, these semantic regions would be segmented into many brushstrokes and it's hard to maintain the original artistic feeling by inflating those brushstrokes (see Fig. 20b). Meanwhile, it would be very time-consuming to adjust the makeup cues for every brushstroke in such semantic regions. Here we select a small region from a van Gogh's painting (F779, Wheatfield with Crows) for high relief generation (see Fig. 20).

8 Other Applications

Once the brushstrokes are extracted, we can re-pose a number of image editing operations to enable interesting paint-aware applications.

8.1 Recoloring Paintings

It is natural to recolor the specified strokes for recomposition. Once the brushstrokes are available, recoloring strokes with a new palette color is becoming as simple as linearly combining N images. Fig. 22 shows recoloring strokes on three paintings respectively, Rosemaling, van Gogh oil painting and Chinese bush painting. As the strokes are extracted, it is easy to separately recolor one or more strokes with different colors. Moreover, Fig. 14 further illustrates the brushstrokes of the van Gogh oil painting, which justifies the efficiency of our brushstroke extraction method.

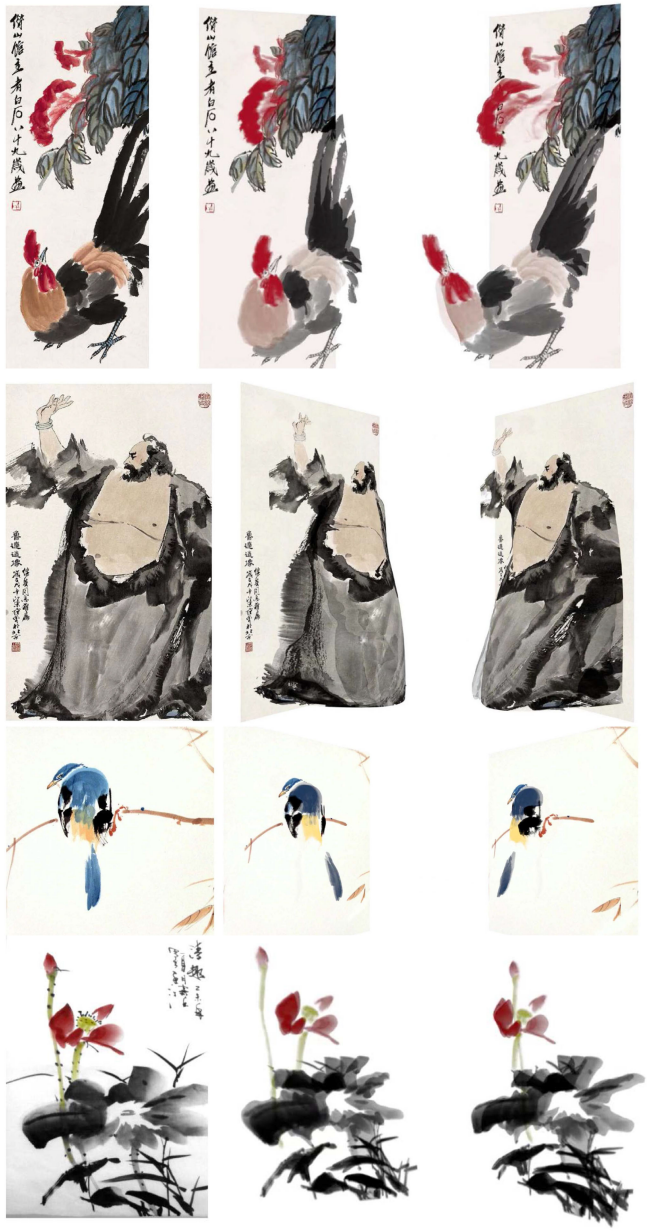


Fig. 21. High relief results from our method (rotated 12 to 45 degree from the original viewing direction).

8.2 Inserting Objects

Fig. 22d shows stroke manipulation through inserting objects for recomposition. One of image synthesis 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 in between two brushstrokes here. The occluded regions of the objects are visible due to the transparency of the brushstrokes. Unlike the traditional image synthesis approaches, our implementation works on the strokes, which both guarantees seamless and keep the transparency of the painting.

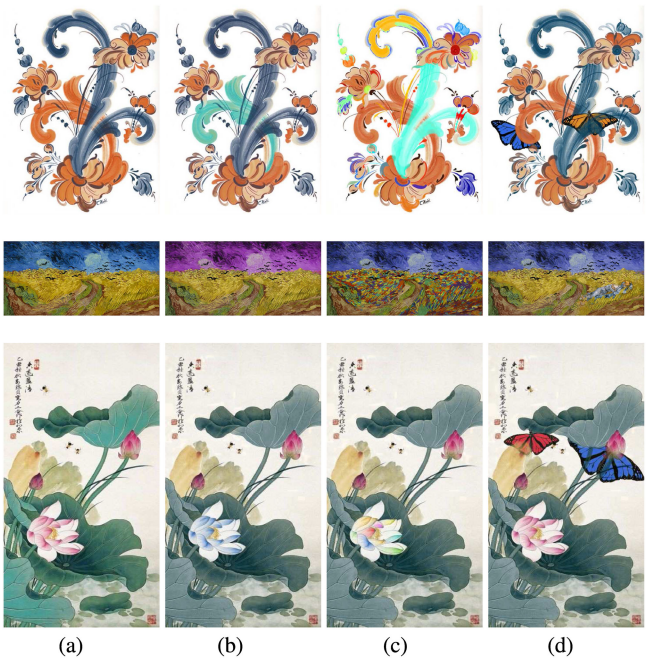


Fig. 22. Column (a) shows 3 kinds of brush paintings, Rosemailing (Europe), van Gogh oil painting, and Chinese painting (East Asia). Column (b) show recoloring one stroke and Column (c) shows recoloring multiple strokes. Column (d) shows inserting objects into these 3 paintings, in which the objects are opaque and are inserted in between brushstrokes.

8.3 Animating Strokes

We also demonstrate our result by moving and rotating the high relief of certain brushstroke to create some simple animation.

9 Conclusions

Relief is an art form part way between a 3D sculpture and 2D painting. We present a new approach for generating a relief from a single brush painting, aiming to preserve the original artistic features of the painting. We particularly consider the overlapped brushstrokes with complex shape profiles and their opacities. These are novel elements that we focus on in this research work. In summary, our contributions include:

1) Extraction of brushstrokes. We develop a novel method to extract brushstrokes based on palette colors analysis and layer decomposition. Comparing with the previous brushstroke extraction methods, our method is capable of extracting overlapped brushstrokes automatically without the prior knowledge of brushstrokes and has a higher accuracy.
2) Relief generation of each brushstroke. We develop a novel method which can generate every brushstroke to a corresponding high-relief surface. By doing so, we preserve the feature of the input painting well.
3) Relief generation from opacity. In contrast to the previous 2D image based methods, our method uses opacity instead of intensity to generate reliefs, which can better preserve the feature of the input painting.
4) Texture-mapping with opacity. Instead of using the raw image as texture, the extracted brushstrokes with opacity value are mapped to the relief surface in our method, which can better show the feature and inter-relations of brushstrokes on relief.
5) Our brushstroke extraction technique has other potential uses. We demonstrate the utility of the decomposed brushstrokes for image editing. See Section 8. Our experiments show that our method is able to produce convincing high-reliefs from a variety of Chinese brush paintings (with human, animal, flower, etc.) and other suitable styles including Rosemaling paintings and oil paintings.

Future Work In the future, we will attempt a wider range of paintings and investigate the issues of their individual layer decomposition and stroke segmentation. Different paintings have different stroke patterns. Understanding the individual rules will improve the success rate for a wider range for paintings. Rational and automatic color separation in the overlap regions is not trivial and will also be studied in the future.

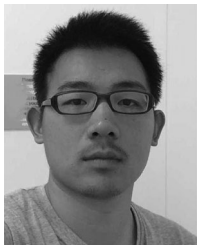
Acknowledgments

We would like to thank the reviewers for many helpful comments. This work was supported in part by the Ministry of Science and Technology (106-2221-E-006-233-MY2, 107-2811-E-006-006- and 107-2221-E-006-196-MY3) and EU H2020 RISE project-AniAge (691215).

References

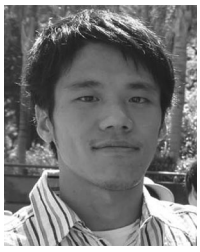
[1] M. Alexa and W. Matusik, "Reliefs as images," *ACM Trans. Graph.*, vol. 29, no. 4, pp. 60–1, 2010.
[2] Z. Benzaid, "Analysis of bas-relief generation techniques," PhD thesis, Dept. of Engineering, Univ. Wisconsin-Milwaukee, Madison, WI, 2017.
[3] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 9, pp. 1124–1137, Sep. 2004.
[4] M. Donoser and H. Bischof, "Efficient maximally stable extremal region (MSER) tracking," in *Proc. IEEE Comput. Society Conf. Comput. Vis. Pattern Recognit.*, 2006, vol. 1, pp. 553–560.
[5] J.-B. Faille, *The works of Vincent van Gogh: His paintings and drawings*, vol. 1070. Alan Wofsy Fine Arts, 1970.
[6] R. B. Girshick, "Simulating chinese brush painting: The parametric hairy brush," in *Proc. ACM SIGGRAPH Posters*, 2004, Art. no. 22.
[7] L. Gomez and D. Karatzas, "A fast hierarchical method for multi-script and arbitrary oriented scene text extraction," *Int. J. Document Anal. Recognit.*, vol. 19, no. 4, pp. 335–349, 2016.
[8] J. Herold and T. F. Stahovich, "A machine learning approach to automatic stroke segmentation," *Comput. Graph.*, vol. 38, pp. 357–364, 2014.
[9] D. Hoiem, A. A. Efros, and M. Hebert, "Automatic photo pop-up," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 577–584, 2005.
[10] D. Hoiem, A. A. Efros, and M. Hebert, "Geometric context from a single image," in *Proc. 10th IEEE Int. Conf. Comput. Vis.*, pp. 654–661, vol. 1, 2005.
[11] D. Hoiem, A. A. Efros, and M. Hebert, "Recovering surface layout from an image," *Int. J. Comput. Vis.*, vol. 75, no. 1, 2007, Art. no. 151.
[12] T. Igarashi, S. Matsuoka, and H. Tanaka, "Teddy: A sketching interface for 3D freeform design," in *Proc. ACM Siggraph Courses*, 2007, Art. no. 21.
[13] H. Kang, S. Lee, and C. K. Chui, "Coherent line drawing," in *Proc. 5th Int. Symp. Non-Photorealistic Animation Rendering*, 2007, pp. 43–50.
[14] O. A. Karpenko and J. F. Hughes, "Smoothsketch: 3D free-form shapes from complex sketches," *ACM Trans. Graph.*, vol. 25, pp. 589–598, 2006.
[15] J. Kerber, A. Tevs, A. Belyaev, R. Zayer, and H.-P. Seidel, "Feature sensitive bas relief generation," in *Proc. IEEE Int. Conf. Shape Modeling Appl.*, 2009, pp. 148–154.
[16] J. Kerber, M. Wang, J. Chang, J. J. Zhang, A. Belyaev, and H.-P. Seidel, "Computer assisted relief generationa survey," *Comput. Graph. Forum*, vol. 31, pp. 2363–2377, 2012.

[17] M. Kolomenkin, G. Leifman, I. Shimshoni, and A. Tal, "Reconstruction of relief objects from line drawings," in *Pro. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 993–1000.
[18] J. Li, L. Yao, E. Hendriks, and J. Z. Wang, "Rhythmic brushstrokes distinguish van gogh from his contemporaries: Findings via automated brushstroke extraction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 6, pp. 1159–1176, Jun. 2012.
[19] P.-L. Lions, E. Rouy, and A. Tourin, "Shape-from-shading, viscosity solutions and edges," *Numerische Mathematik*, vol. 64, no. 1, pp. 323–353, 1993.
[20] B. Liu, S. Gould, and D. Koller, "Single image depth estimation from predicted semantic labels," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 1253–1260.
[21] A. V. Loon, "Color changes and chemical reactivity in seventeenth-century oil paintings," *AMOLF. FOM Institute for Atomic and Molecular Physics*, 2008.
[22] J. Malik, "Interpreting line drawings of curved objects," *Int. Jo. Comput. Vis.*, vol. 1, no. 1, pp. 73–103, 1987.
[23] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," *Image Vis. Comput.*, vol. 22, no. 10, pp. 761–767, 2004.
[24] J. McCann and N. Pollard, "Local layering," *ACM Trans. Graph.*, vol. 28, no. 3, 2009, Art. no. 84.
[25] J. McCann and N. S. Pollard, "Soft stacking," *Comput. Graph. Forum*, vol. 31, pp. 469–478, 2012.
[26] A. Nealen, T. Igarashi, O. Sorkine, and M. Alexa, "Fibermesh: Designing freeform surfaces with 3D curves," *ACM Trans. Graph.*, vol. 26, no. 3, 2007, Art. no. 41.
[27] D. Nistér and H. Stewénius, "Linear time maximally stable extremal regions," in *Proc. 10th Eur. Conf. Comput. Vis.*, 2008, pp. 183–196.
[28] B. M. Oh, F. Durand, and M. Chen, "Image-based modeling and photo editing," US Patent 7,199,793, Apr. 3, 2007.
[29] T. Porter and T. Duff, "Compositing digital images," *ACM Siggraph Comput. Graph.*, vol. 18, pp. 253–259, 1984.
[30] E. Prados and O. Faugeras, "Unifying approaches and removing unrealistic assumptions in shape from shading: Mathematics can help," in *Proc. Eur. Conf. Comput. Vis.*, 2004, pp. 141–154.
[31] E. Prados and O. D. Faugeras, "Perspective shape from shading" and viscosity solutions," in *Proc. 9th IEEE Int. Conf. Comput. Vis.*, 2003, vol. 3, Art. no. 826.
[32] C. Richardt, J. Lopez-Moreno, A. Bousseau, M. Agrawala, and G. Drettakis, "Vectorising bitmaps into semi-transparent gradient layers," *Comput. Graph. Forum*, vol. 33, pp. 11–19, 2014.
[33] A. Saxena, M. Sun, and A. Y. Ng, "Make3d: Learning 3D scene structure from a single still image," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 5, pp. 824–840, May 2009.
[34] J. A. Sethian, *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*, vol. 3. Cambridge, U.K.: Cambridge Univ. Press, 1999.
[35] R. Smith and E. J. Lloyd, "Art school," Dorling Kindersley Ltd, Inc., 1997.
[36] D. Sýkora, L. Kavan, M. Čadík, O. Jamriška, A. Jacobson, B. Whited, M. Simmons, and O. Sorkine-Hornung, Ink-and-ray: Bas-relief meshes for adding global illumination effects to hand-drawn characters," *ACM Trans. Graph.*, vol. 33, no. 2, 2014, Art. no. 16.
[37] J. Tan, J.-M. Lien, and Y. Gingold, "Decomposing images into layers via RGB-space geometry," *ACM Trans. Graph.*, vol. 36, no. 1, 2016, Art. no. 7.
[38] P. A. Varley and R. R. Martin, "Estimating depth from line drawing," in *Proc. 7th ACM Symp. Solid Modeling Appl.*, 2002, pp. 180–191.
[39] T. Weyrich, J. Deng, C. Barnes, S. Rusinkiewicz, and A. Finkelstein, "Digital bas-relief from 3D scenes," in *Proc. ACM Trans. Graph.*, 2007, vol. 26, Art. no. 32.
[40] J. Wu, R. R. Martin, P. L. Rosin, X.-F. Sun, F. C. Langbein, Y.-K. Lai, A. D. Marshall, and Y.-H. Liu, "Making bas-reliefs from photographs of human faces," *Comput.-Aided Des.*, vol. 45, no. 3, pp. 671–682, 2013.
[41] T.-P. Wu, J. Sun, C.-K. Tang, and H.-Y. Shum, "Interactive normal reconstruction from a single image," *ACM Trans. Graph.*, vol. 27, 2008, Art. no. 119.
[42] S. Xu, Y. Xu, S. B. Kang, D. H. Salesin, Y. Pan, and H.-Y. Shum, "Animating chinese paintings through stroke-based decomposition," *ACM Trans. Graph.*, vol. 25, no. 2, pp. 239–267, 2006.
[43] Z. Xu and J. Sun, "Image inpainting by patch propagation using patch sparsity," *IEEE Trans. Image Process.*, vol. 19, no. 5, pp. 1153–1165, May 2010.
[44] C.-K. Yeh, S.-Y. Huang, P. K. Jayaraman, C.-W. Fu, and T.-Y. Lee, "Interactive high-relief reconstruction for organic and double-sided objects from a photo," *IEEE Trans. Vis. Comput. Graph.*, vol. 23, no. 7, pp. 1796–1808, Jul. 2017.
[45] Q. Zeng, R. R. Martin, L. Wang, J. A. Quinn, Y. Sun, and C. Tu, "Region-based bas-relief generation from a single image," *Graphical Models*, vol. 76, no. 3, pp. 140–151, 2014.
[46] L. Zhang, G. Dugas-Phocion, J.-S. Samson, and S. M. Seitz, "Single-view modelling of free-form scenes," *Comput. Animation Virtual Worlds*, vol. 13, no. 4, pp. 225–235, 2002.



Yunfei Fu received the BS degree from the Department of Electrical Engineering and Computer Science, Dalian University of Technology, in 2013. He is currently working toward the PhD degree at the National Centre for Computer Animation, Bournemouth University, Poole, United Kingdom. His research interests include computer graphics, nonphotorealistic rendering, and computer vision.

Hongchuan Yu received the PhD degree in computer vision from the Institute of Intelligent Machine, Chinese Academy of Sciences, Beijing, PRC, in 2000. He is currently a lecturer with the National Centre for Computer Animation, Bournemouth University, Poole, United Kingdom. His research interests include Geometry modeling and rendering, face recognition and expression synthesis, and image processing.



Chih-Kuo Yeh received the BS degree from the Department of Information Engineering and Computer Science, Feng Chia University, in 2005, the MS degree from the Institute of Bioinformatics, National Chiao Tung University, in 2007, and the PhD degree from the Department of Computer Science and Information Engineering, National Cheng-Kung University, Taiwan, in 2015. He is currently a postdoctoral researcher with National Cheng-Kung University. His research interests include scientific visualization, computer animation, and computer graphics.

Jianjun Zhang is currently a professor of computer graphics with the National Centre for Computer Animation, Bournemouth University, United Kingdom, where leads the Computer Animation Research Centre. He is also a cofounder of the UKs Centre for Digital Entertainment. His research focuses on 3D virtual human modeling, animation and simulation, including geometric modeling, rigging and skinning, motion synthesis, deformation, and physics-based simulation.

Tong-Yee Lee received the PhD degree in computer engineering from Washington State University, Pullman, in May 1995. He is currently a chair professor with the Department of Computer Science and Information Engineering, National ChengKung University, Tainan, Taiwan, ROC. He leads the Computer Graphics Group, Visual System Laboratory, National Cheng-Kung University (http://graphics.csie.ncku.edu.tw/). His current research interests include computer graphics, nonphotorealistic rendering, medical visualization, virtual reality, and media resizing. He is a senior member of the IEEE Computer Society and a member of the ACM.