

中图法分类号: TP391.41 文献标识码: A 文章编号: 1006-8961(2023)12-3825-13

论文引用格式: Rao S J, Qian W H and Zhang J B. 2023. Image style transfer via curved stroke rendering. Journal of Image and Graphics, 28(12): 3825-3837(饶师瑾, 钱文华, 张结宝. 2023. 曲线笔触渲染的图像风格迁移. 中国图象图形学报, 28(12): 3825-3837)[DOI: 10.11834/jig.221150]

曲线笔触渲染的图像风格迁移

饶师瑾, 钱文华*, 张结宝

云南大学信息学院, 昆明 650504

摘要: 目的 针对 GANILLA、Paint Transformer、StrokeNet 等已有的风格迁移算法存在生成图像笔触丢失、线条灵活度低以及训练时间长等问题, 提出一种基于曲线笔触渲染的图像风格迁移算法。方法 首先按照自定义的超像素数量将图像前景分割为小区域的子图像, 保留更多图像细节, 背景分割为较大区域的子图像, 再对分割后的每个子区域选取控制点, 采用 Bezier 方程对控制点进行多尺度笔触生成, 最后采用风格迁移算法将渲染后的图像与风格图像进行风格迁移。结果 与 AST(arbitrary style transfer)方法相比, 本文方法在欺骗率指标上提升了 0.13, 测试者欺骗率提升了 0.13。与 Paint Transformer 等基于笔触渲染的算法对比, 本文能够在纹理丰富的前景区域生成细粒度笔触, 在背景区域生成粗粒度笔触, 保存更多的图像细节。结论 与 GANILLA、AdaIN(adaptive instance normalization)等风格迁移算法相比, 本文采用图像分割算法取点生成笔触参数, 无需训练, 不仅提高了算法效率, 而且生成的多风格图像保留风格化图像的笔触绘制痕迹, 图像色彩鲜明。

关键词: 非真实感渲染; 风格迁移; 笔触渲染(SBR); Bezier 曲线; 超像素分割

Image style transfer via curved stroke rendering

Rao Shijin, Qian Wenhua*, Zhang Jiebao

School of Information Science and Engineering, Yunnan University, Kunming 650504, China

Abstract: Objective The goal of the image style transfer algorithm is to render the content of one image with the style of another image. Image style transfer methods can be divided into traditional and neural style transfer methods. Traditional style transfer methods can be broadly classified to stroke-based rendering (SBR) and image analogy (IA). SBR simulates human drawings with different sizes of strokes. Meanwhile, the main idea of IA is as follows: given a pair of images A (unprocessed source image) and A' (processed image) and the unprocessed image B , the processed image B' is obtained by processing B in the same way as A to A' . Meanwhile, neural style transfer methods can be categorized into slow image reconstruction methods based on online image optimization and fast image reconstruction methods based on offline model optimization. Slow image reconstruction methods optimize the image in the pixel space and minimize the objective function via gradient descent. Using a random noise as the starting image, the pixel values of the noise images are iteratively changed to obtain a target result image. Given that each reconstruction result requires many iterative optimizations in the pixel space, this approach consumes much time and computational resources and requires a high time overhead. In order to speed up this process, fast image reconstruction methods are proposed to train the network in advance in a data-driven manner using a large amount of data. Given an input, the trained network only needs one forward transmission to output a style

收稿日期: 2022-12-22; 修回日期: 2023-03-01; 预印本日期: 2023-03-08

* 通信作者: 钱文华 whqian@ynu.edu.cn

基金项目: 国家自然科学基金项目(62162065); 云南省科技厅基金项目(202105AF150011)

Supported by: National Natural Science Foundation of China (62162065)

transfer image. In recent years, the seminal works on style transfer have focused on building a neural network that can effectively extract the content and style features of an image and then combine these features to generate highly realistic images. However, building a model for each style is inefficient and requires much labor and time resources. One example of this model is the neural style transfer (NST) algorithm, which aims at transferring the texture of the style image to the content image and optimizing the noise at the pixel level step by step. However, hand-painted paintings comprise different strokes that are made using different brush sizes and textures. Compared with human paintings, the NST algorithm only generates photo-realistic imageries and ignores paint strokes or stipples. Given that the existing style transfer algorithms, such as Ganilla and Paint Transformer, suffer from loss of brush strokes and poor stroke flexibility, we propose a novel style transfer algorithm to quickly recreate the content of one image with curved strokes and then transfer another style to the re-rendered image. The images generated using our method resemble those made by humans. **Method** First, we segment the content image into subregions with different scales via content mask according to the customized number of super pixels. Given that we do not pay attention to the background, we segment the image background into small subregions. To preserve large amounts of details, we save the image foreground as much as possible via segmentation into small subregions. The segmentations for the image foreground are twice greater than those for the image background. For each subregion, four control points are selected in the convex hull of a subregion, and then the Bezier equation is used to generate thick strokes in the background and thin strokes in the foreground. The image rendered with strokes is then stylized with the style image by using the style transfer algorithm to generate a stylized image that retains the stroke traces. **Result** Compared with the arbitrary style transfer (AST) and Kotovenko's method, the deception rate of the proposed method is increased by 0.13 and 0.04, respectively, while its human deception rate is increased by 0.13 and 0.01. Compared with Paint Transformer and other stroke-based rendering algorithms, our proposed method can generate thin strokes in the texture-rich foreground region and thick strokes in the background, thus preserving large amounts of image details. **Conclusion** Unlike whitening and coloring transforms (WCT), AdaIN, and other style transfer algorithms, the proposed method uses an image segmentation algorithm to generate stroke parameters without training, thus improving efficiency and generating multi-style images that preserve the stroke drawing traces of stylized images with vivid colors.

Key words: non-photorealistic rendering; style transfer; stroke-based rendering(SBR); Bezier curve; superpixel segmentation

0 引言

图像风格迁移(style transfer)可以将一幅图像的风格迁移到另一幅图像(Gatys等,2015),模拟艺术化绘制方式,创造出更多类似大师手绘的作品。传统风格迁移算法(Hertzmann等,2001)通常采用像素到像素之间的映射(刘哲良等,2019),或连续优化数字图像中的每个像素点,生成风格迁移图像。逐像素生成风格迁移图像的方法与艺术家绘画的方法不同,绘画需要按照一定的笔画顺序,逐笔绘制出图像的整体与局部特征。然而,艺术家在创作不同类型的作品时需要使用粗细、形状不同的画笔描绘图像中的背景和细节,不同的画笔、颜料以及绘画方式在纸张上会表现出不同的笔触痕迹,例如,墨汁在宣纸上作画会有晕染的效果,而用丙烯颜料作画,颜料之间的相互覆盖会在纸张上形成堆叠效果,产生层次感,这些复杂的笔触效果采用逐像素点优化的

方式难以实现。

为了模拟人类绘画的真实过程,Hertzmann(2003)提出了基于笔触渲染(stroke-based rendering, SBR)的非真实感图像渲染算法。基于笔触渲染的风格迁移算法主要包括基于强化学习的算法(Xie等,2013;Ganin等,2018;Zheng等,2023;Huang等,2019)和基于优化的算法(Hertzmann,1998;Liu等,2021;Zou等,2021)。强化学习算法在智能体与环境交互的过程中,通过长期激励,鼓励智能体学习如何将目标图像分解为多个笔触,再迭代地将笔触重新渲染到画布上,虽然强化学习算法可以较快渲染出图像,但是需要耗费很长时间训练智能体,在缺少学习样本的情况下,很难训练出稳定的智能体决策每一个笔触的位置和颜色等参数,此外,强化学习还需要考虑笔触之间的组合关系和覆盖关系,若两个笔触之间相互覆盖,强化学习方法生成的图像会产生十分明显的叠加效果和笔触痕迹。

图1(a)展示了强化学习(Huang等,2019)算法

生成的结果图像,图中人物面部笔触颜色差异大,边界明显,可以明显看到笔触痕迹。基于优化的笔触渲染算法通常采用贪心搜索策略,在较大的搜索空间中寻求满足条件的笔触参数集合。基于优化的笔触渲染算法主要分为两类,试错算法和Voronoi算法(Hertzmann, 2003)。试错算法先将目标图像分割,再采用其他图形匹配目标图像的分割结果,例如,三角形、圆形、矩形等,如果当前匹配的形状放置到画布上能够使能量函数减小,就将该形状渲染到画布上;反之,则进行下一轮的预测,匹配新的图形。

图1(b)展示了试错算法的结果图像,Song等人(1998)采用Graph-cuts算法对图像进行分割,再通过不规则图形拼凑图像,这类算法需要大量的时间不断试错。Voronoi算法将目标图像分割成一系列子区域,在每个子区域中确定笔触的中心位置,然后将笔触渲染到图像中,再参考目标图像优化笔触的中心位置,笔触中心位置的优化过程与笔触密度有关,优化时间随着笔触密度的增加而延长。

图1(c)展示了Voronoi算法的结果图像,Zou等人(2021)采用由低分辨率到高分辨率的渐进渲染策略,在每一个渲染阶段,将画布分割为若干个相交的子区域,每一个子区域都需要进行相同次数的梯度下降并对区域中的所有笔触进行优化,子区域中笔触越多,程序运行时间越长。在真实的艺术作品中,艺术家通常采用弧形或不规则的笔触描绘图像色彩连续的区域,然而目前基于神经网络的笔触渲染算法通过对初始笔触进行仿射变换,使笔触满足目标图像分布的方法生成图像,与艺术家的绘画方式大相径庭,因为仿射变换产生曲线笔触较困难,并且采用单一的直线笔触绘画显得画作十分刻板,缺乏艺术感,无法绘制出图像中的细小纹理。

为了增强笔触的灵活度,保留风格迁移图像的

笔触痕迹,降低程序运行时间,本文提出一种基于曲线笔触渲染的风格迁移算法,首先将图像前景和背景进行多尺度分割,再根据分割后的子图像选取多个节点,通过节点坐标,采用三次Bezier曲线生成虚拟曲线笔触;其次,将曲线笔触从离散域变换到像素域,渲染到画布上;最后,将渲染后的图像与风格图像进行风格迁移。实验证明,本文提出的采用多尺度曲线笔触渲染图像的方法,图像中的细小纹理丢失更少,前景与背景之间的界限更清晰,生成的风格化图像色彩艳丽,且保留了真实绘画的笔触痕迹。

1 相关工作

1.1 风格迁移

单风格迁移模型仅能生成特定风格的图像。Gatys等人(2016)提出采用卷积神经网络(convolutional neural network, CNN)最小化随机噪声图像与风格图像和内容图像之间的风格损失和内容损失,将风格图像的风格与内容图像的语义信息结合,生成一幅新的风格图像,但生成图像内容信息丢失严重。Johnson等人(2016)在Gatys等人(2016)的工作基础上采用感知损失函数训练前馈神经网络,减少内容图像的细节丢失,生成与风格图像接近、但内容结构保留更完整的风格化图像。Lin等人(2021)提出一种基于前馈网络的LapStyle(Laplacian pyramid network)算法,采用拉普拉斯算法提取内容图像的轮廓,保存细节并在渐进过程中合成高清风格化图像。单风格迁移模型需要为特定风格重新搭建并训练模型,缺乏灵活性,而多风格迁移模型仅需一次训练即可以适用于多种风格迁移。An等人(2021)提出基于可逆神经流的PFN(projection flow network)网络,既可实现图像的多风格迁移,又采用神经网络的正向与逆向流动机制,解决了大多数风格迁移网络中存在的内容泄露现象,实现无损风格迁移。与多风格迁移模型相比,任意风格迁移模型适用于任意一种风格。Li等人(2017)提出基于白化和着色变换的通用风格迁移算法,无需对每种风格单独训练即可生成高质量风格化图像。Wang等人(2020)将正交随机噪声引入到Li等人(2017)的工作中,增加任意风格迁移结果的多样性。谢斌等人(2020)提出基于相关对齐的图像风格纹理提取方法,可以有效抑制迁移过程中噪声的产生,生成的风格图像纹理



图1 各类算法的绘画式重建效果

Fig. 1 Image to painting results of various algorithms

((a) reinforcement learning algorithm;

(b) trial-and-error algorithm; (c) Voronoi algorithm)

分布更加均匀,并且图像生成效率更高。

1.2 超像素分割

超像素分割根据像素的颜色、纹理和位置等信息将视觉上相似的像素点分割为一个子区域,每个子区域表示一个超像素。超像素分割算法分为基于图论的算法和基于梯度上升的算法。基于图论的算法将图像映射为一幅图,图节点表示图像中所有的像素点,节点之间相连的边代表像素点之间的相邻关系,采用最大流、最小割算法最小化能量函数,生成超像素。Zhang等人(2019)在风格迁移任务中,将图像的内容特征和风格特征作为图节点,通过Graph-cuts算法对图像内容特征和风格特征进行语义和空间上的匹配,实现多模态风格迁移。基于图论的超像素分割算法,计算量随着图像尺寸的增大而增加,且能量函数的构造以及能量函数最小化问题都属于NP-hard问题。基于梯度上升的超像素分割算法需要先初始化聚类中心,再对所有聚类中心迭代运用梯度上升算法更新聚类中心,直到每一个聚类中心都满足收敛条件为止。Achanta等人(2012)提出的简单线性迭代聚类(simple linear iterative clustering, SLIC)算法,可以快速生成边缘贴合度高、分割大小均匀的超像素,提高了超像素分割算法的效率,但是SLIC算法仅适用于分割规则图像,为了能够在非规则区域内同样达到SLIC算法的效果,Irving(2016)提出可以在感兴趣区域分割出均匀超像素的maskSLIC算法,该算法优化了SLIC算法的初始聚类中心生成方法,使聚类中心能够均匀地散布在感兴趣区域内。超像素分割在医学图像分割(Amami等,2019)、目标检测(Ghariba等,2022)和风格迁移(Zhang等,2019)上都有很广泛的应用。本文采用maskSLIC算法对图像分割,获取图像分割后的标签,作为曲线笔触生成的预处理步骤。

1.3 笔触渲染算法

笔触渲染是计算机模拟人类绘画的方式,对自然图像进行艺术化渲染,生成非真实感图像的过程(Hertzmann,1998)。基于优化的笔触渲染算法采用直线笔触(Zou等,2021;Liu等,2021)或其他形状的图形绘制图像(Hertzmann,1998),Paint Transformer模型(Liu等,2021)通过预测的笔触参数,变换初始油画笔触的形状、颜色和位置等,生成直线笔触,仅由直线笔触组成的绘画作品缺乏灵活性。与Paint Transformer相比,Kotovenko等人(2021)提出的参数

化笔触渲染算法可以生成更加细小的笔触,但是由于受控制点数量、位置等因素的影响,该模型生成的笔触更加趋近于直线笔触,并且整幅图像均采用大小相同的笔触进行渲染,导致生成的图像前景轮廓不清晰,细节丢失严重。

2 本文方法

2.1 整体框架

图2展示了本文方法的整体框架,首先通过maskSLIC超像素分割算法将内容图像的前景、背景分割为不同尺度的子图像,在每一个子图像中分别取出4个节点,作为曲线的控制点,根据实验结果,选取4个节点,可以在保证算法效率的前提下,生成弯曲程度更佳的曲线笔触,采用3次Bezier曲线方程对4个节点进行拟合,生成曲线笔触,将笔触渲染到空画布上,再通过风格损失和内容损失优化笔触的颜色、位置、形状和颜色等参数,最后将绘制出来的内容图像与风格图像进行风格迁移。

2.2 基于掩膜的多尺度超像素分割

超像素分割算法先将图像从RGB空间转换到CIELAB(international commission on illumination lab color space)空间,因为CIELAB空间色域更广,对图像的原生色彩不会造成影响,再计算每个像素点在CIELAB空间中的色彩距离和坐标距离,将颜色相近、位置相邻的像素划分到同一个区域。采用Achanta等人(2012)提出的像素点相似度衡量标准,计算子图像中每一个像素点与其他像素点的颜色差异和位置距离。具体为

$$d_{\text{color}} = \sqrt{(l_i - l_j)^2 + (a_i - a_j)^2 + (b_i - b_j)^2} \quad (1)$$

$$d_{\text{location}} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (2)$$

$$f_{\text{distance}} = d_{\text{color}} + \beta d_{\text{location}} \quad (3)$$

式中, d_{color} 表示像素点在CIELAB空间的色彩差异, l_i 、 a_i 、 b_i 分别表示像素点*i*在CIELAB空间中的明暗强度、红绿分量和蓝黄分量, d_{location} 表示像素点的空间距离, x 和 y 分别表示坐标点的横、纵坐标, f_{distance} 表示CIELAB色彩空间距离和归一化后的空间距离之和, f_{distance} 的值越大表示两个像素越相似, β 表示空间距离 d_{location} 的权重。由于maskSLIC算法可以自定义超像素的数量,并将图像分割为大小均匀的子图像,所以本文采用maskSLC算法对图像进行不同

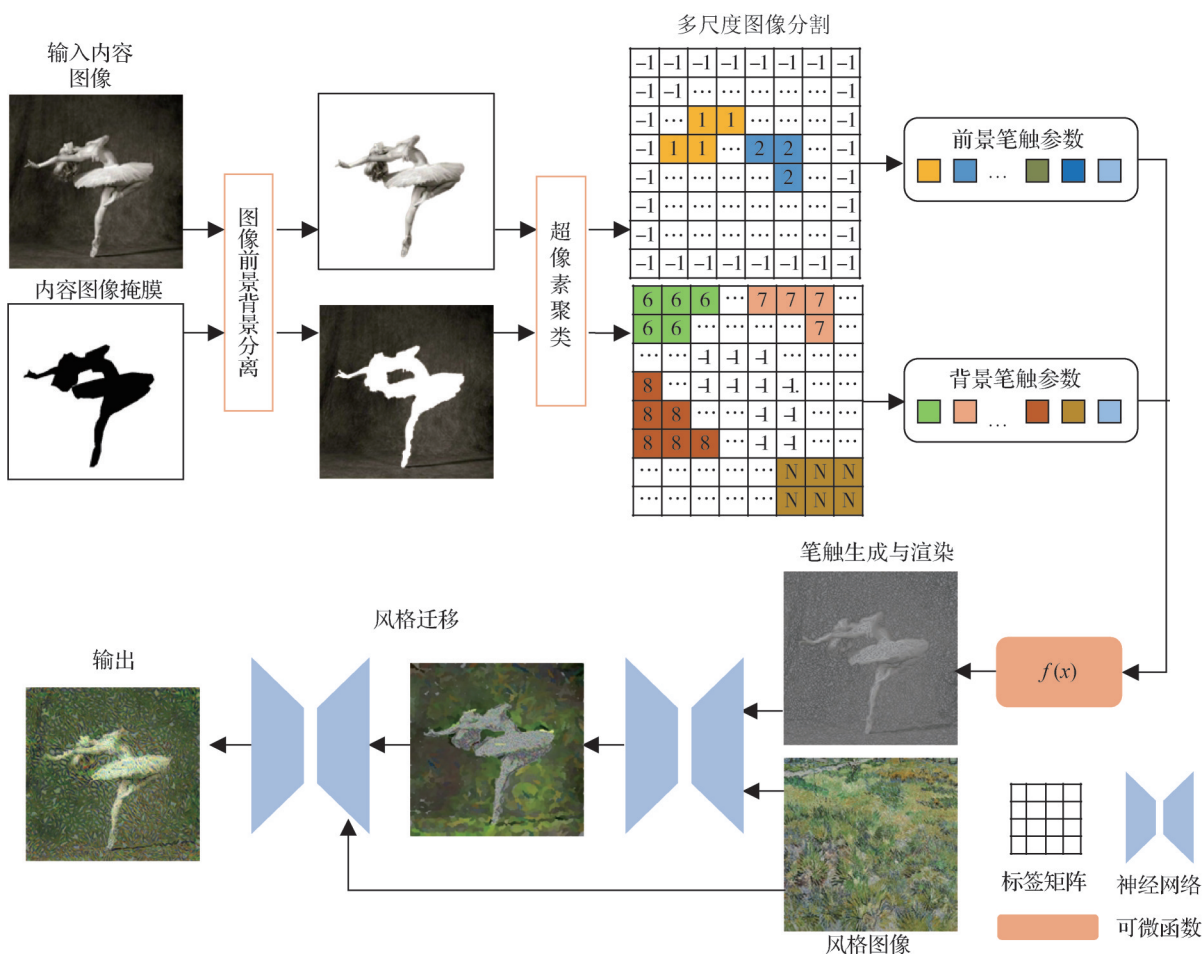


图2 整体框架

Fig. 2 Overall framework

尺度的分割,生成不同尺度的笔触,分别渲染图像前景和背景。maskSLIC算法共分为两个步骤,具体如下

1)在掩膜区域中初始化聚类中心。采用 S_k 表示聚类中心集合, M 表示掩膜区域内点的集合, $\hat{M}$ 表示掩膜区域外点的集合, L 表示掩膜区域外的点与聚类中心的集合, $L = \hat{M} \cup S_k$ 。通过欧几里德距离变换公式初始化掩膜区域内的聚类中心,具体为

$$D(x) = \min_{y \in L} \sqrt{\sum_i^n (x_i - y_i)^2} \quad (4)$$

式中, $D(x)$ 表示 M 中的像素点 x 与 L 中的像素点 y 之间的距离变换, n 表示图像的维度。

2)为了有效减少聚类中心优化的迭代次数,加快图像分割算法的效率,采用k-means++算法迭代优化聚类中心的位置,具体如算法1所示。

算法1:基于掩膜的超像素分割算法。

输入:图像 I ,超像素个数 N ,图像高度 H ,图像宽度 W ,图像掩膜 M ;

输出:图像的标签集合

$$L_p = \{L_p^{(1)}, L_p^{(2)}, L_p^{(3)}, \dots, L_p^{(H \times W)}\}$$

1) 初始化 $S_k = \{\}$, $\hat{M} = 1 - M$, $L = \hat{M}$;

2) For i in range $(H \times W)$;

3) $L_p^{(i)} = -1$;

4) 初始化掩膜内每个像素点的距离

$$f_{\text{Distance}} = \infty;$$

5) End for

6) For i in range (N) ;

7) 根据式(4)计算 M 中的点与 L 中的点之间的欧氏距离;

8) 取 $D(x)$ 最大的点 x 为聚类中心,令 $S_k^{(i)} = x$,将 $S_k^{(i)}$ 添加到 S_k ;

9) $L = \hat{M} \cup S_k$;

10) End For;

11) $S_k = \{S_k^{(1)}, S_k^{(2)}, S_k^{(3)}, \dots, S_k^{(N)}\}$;

12) For $S_k^{(i)}$ in S_k ;

- 13) 对于 $S_k^{(i)} 3 \times 3$ 邻域内的每一个像素点 p ;
- 14) 计算点 p 与聚类中心 $S_k^{(i)}$ 的距离 $f_{\text{distance}}(p)$;
- 15) If $f_{\text{distance}}(p) < f_{\text{Distance}}$;
- 16) 令 $f_{\text{distance}}(p) = f_{\text{Distance}}$;
- 17) 令 $L_p^{(i)} = k$;
- 18) End If;
- 19) 计算新的聚类中心;
- 20) End For;
- 21) Return $L_p = \{L_p^{(1)}, L_p^{(2)}, L_p^{(3)}, \dots, L_p^{(H \times W)}\}$ 。

通过 maskSLIC 算法,可以有效分割图像掩膜区域内的像素点,分别获取图像的前景背景标签,分割时可以通过改变掩膜区域的分割数量控制子区域的大小,从而调整后续生成笔触的大小,分割数量越少,分割后的子区域就越大,生成的笔触也就越大,图像丢失的细节就越多,反之,分割数量越多,分割子区域越小,生成笔触也就越小,能够更好地描绘出图像中的细节。

2.3 笔触生成和渲染

Kotovenko 等人(2021)采用起点、控制点和终点共3个节点生成曲线笔触,由于3个节点处于一条直线上,所以生成的笔触弯曲程度不明显,更接近直线笔触。为了模拟真实艺术作品中的绘画笔触,生成细长、弯曲程度显著的曲线笔触,本文采用4个节点控制曲线笔触的生成,过多的节点和太复杂的曲线生成方式会大幅度降低笔触参数的生成效率。

图3为本文采用的曲线笔触参数,包括形状参数 p_0, p_1, p_2, p_3 、颜色参数 c 、宽度参数 w 和位置参数 x, y ,共8个参数,其中,4个形状参数分别表示曲线笔触的起点 p_0 、第1个控制点 p_1 、第2个控制点 p_2 和终点 p_3 ,位置参数 x 表示笔触的横坐标, y 表示笔触

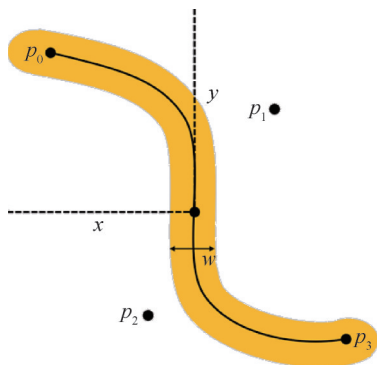


图3 曲线笔触参数

Fig. 3 The parameters of curved stroke

的纵坐标。

形状参数的取点方式决定曲线笔触的整体形态以及弯曲程度,根据2.2节中获取的图像前景背景标签,依次获取相同标签的像素点,在相同标签的像素点范围内计算其对应的凸包集合,再计算凸包集合中每个坐标点之间的距离,取相距最远的两个点分别作为曲线的起点 p_0 和终点 p_3 ,将两个点之间的距离三等分,取第1个三等分点处,子区域的第1个边缘点作为第1个控制点 p_1 ,取第2个三等分点处,子区域的最后一个边缘点作为第2个控制点 p_2 。算法2描述了由图像标签生成曲线笔触参数的详细过程。

算法2:笔触参数生成。

输入:内容图像的标签集合

$L_p = \{L_p^{(1)}, L_p^{(2)}, L_p^{(3)}, \dots, L_p^{(H \times W)}\}$,超像素个数 N 。

输出:曲线参数集合

$B = \{B^{(1)}, B^{(2)}, B^{(3)}, \dots, B^{(N)}\}$ 。

- 1) 初始化笔触参数为空;
 - 2) 获取最大标签值 $\max(L_p)$;
 - 3) For i in range($\max(L_p) + 1$);
 - 4) 取标签值为 i 的全体元素坐标点 P_k ;
 - 5) 计算 P_k 的凸包集合;
 - 6) 计算凸包集合中所有点的坐标距离;
 - 7) 取凸包中相距最远的两个点作为起点 p_0 和终点 p_3 ;
 - 8) 计算点 p_0 和点 p_3 的直线距离 d ;
 - 9) 将 d 三等分,4个隔断点分别为 p_0, d_1, d_2, p_3 ;
 - 10) For j in P_k ;
 - 11) 取横坐标为 d_1 的第一个点 j 为 p_1 ;
 - 12) 取横坐标为 d_2 的最后一个点 j 为 p_2 ;
 - 13) End For;
 - 14) 计算与 p_0 和 p_3 正交的两个坐标点;
 - 15) 计算两个正交点之间的距离作为曲线的宽度 w ;
 - 16) 取 P_k 中 RGB 均值为颜色参数 c ;
 - 17) 取 P_k 中所有横坐标的均值为 x ;
 - 18) 取 P_k 中所有纵坐标的均值为 y ;
 - 19) $B^{(i)} = \{p_0, p_1, p_2, p_3, w, c, x, y\}$;
 - 20) End For;
 - 21) Return $B = \{B^{(1)}, B^{(2)}, B^{(3)}, \dots, B^{(N)}\}$ 。
- 为了能够在子区域的形状弯曲度比较大的情

况下,生成弯曲程度较大的曲线笔触,子区域的形状贴近矩形时,生成弯曲幅度小的笔触,本文在选取控制点时尽量使两个控制点相距较远,图4表示了子区域中取点生成曲线笔触,并将笔触渲染在画布上的部分结果,图中 p_0, p_1, p_2, p_3 分别表示曲线笔触的起点、第1个控制点、第2个控制点和终点。

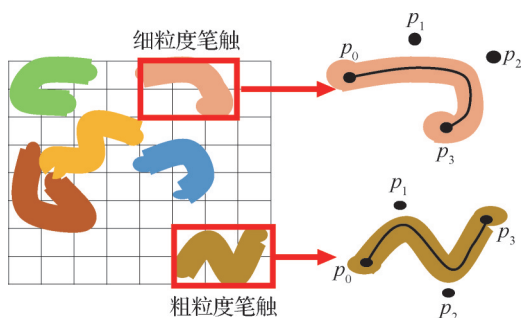


图4 曲线笔触渲染方法

Fig. 4 Curved stroke rendering method

曲线生成方程为

$$\begin{aligned} \text{Stroke}(t) = & (1-t)^3 p_0 + 3(1-t)t p_1 + \\ & 3(1-t)t^2 p_2 + t^3 p_3 \\ & 0 \leq t \leq 1 \end{aligned} \quad (5)$$

式中, t 表示节点顺序, $\text{Stoke}(\cdot)$ 表示笔触位置参数计算。

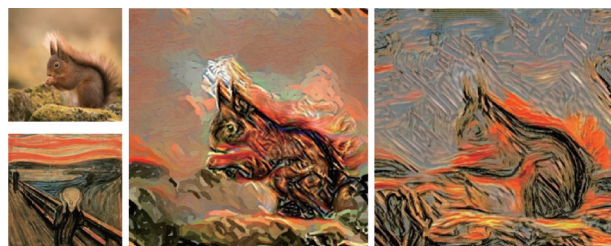
目前,主流的SBR(stroke-based rendering)算法通过神经网络将离散参数转化为矢量笔触图像,再将笔触映射到像素域,训练神经网络不仅需要花费大量的时间,而且需要占用大量的内存,所以本文采用简单的可微函数将笔触集合转换为画布上的像素值,具体为

$$I_{\text{output}} = \text{Renderer}(\text{param}, C) \quad (6)$$

式中, $\text{Renderer}(\cdot)$ 表示渲染计算, I_{output} 和 C 分别表示输出图像和输入图像, param 表示笔触参数集合,渲染器接收笔触参数集合将所有笔触渲染到空画布上。

图5展示了本文方法和Kotovenko等人(2021)方法风格化结果。图5(a)表示输入图像,由上至下分别表示内容图像和风格图像;图5(b)由本文方法生成,共3 000条笔触;图5(c)由Kotovenko等人(2021)的方法生成,共5 000条笔触。本文方法在笔触数量更少的情况下,生成的图像中松鼠的眼睛和耳朵轮廓更清晰,腿部形态更明显,背景中灰色部分

更少,由于笔触痕迹明显、柔和、色彩鲜艳,因此图像整体视觉效果更佳。



(a) 输入图像 (b) 本文方法 (c) Kotovenko等人(2021)

图5 本文方法与Kotovenko等人(2021)方法的比较结果

Fig. 5 Comparison of results between Kotovenko et al. (2021) method and ours ((a) input images; (b) ours; (c) Kotovenko et al. (2021))

2.4 损失函数

本文算法主要分为两个阶段,笔触渲染阶段和风格迁移阶段。为了能够增强合成图像的纹理信息,保留内容图像的结构特征,两个阶段均采用了风格损失和内容损失,在渲染阶段,采用曲率损失调整曲线控制点的位置,增强曲线弯曲度。

风格迁移损失可以在图像渲染阶段,逐渐使每一条笔触颜色趋近风格图像的色彩,风格迁移阶段使渲染后的图像增强风格化图像的纹理信息,计算为

$$G_{\text{render}}^l = \sum_{i=0}^L \phi_l(I_{\text{render}}) (\phi_l(I_{\text{render}}))^T \quad (7)$$

$$E_l = \left(\sum_{i=0}^L (G_{\text{render}}^l - G_{\text{style}}^l) \right) / N_l^2 M_l^2 \quad (8)$$

$$L_{\text{sty}} = \sum_{l=0}^L w_l E_l \quad (9)$$

式中, I_{render} 表示渲染生成的内容图像, ϕ_l 表示图像第 l 层的特征, G_{render}^l 表示生成图像的Gram矩阵, G_{style}^l 表示风格图像的Gram矩阵, $N_l^2 M_l^2$ 是第 l 层的归一化常数, E_l 是第 l 层的风格损失, w_l 表示风格损失的权重, L_{sty} 表示总风格损失。

内容损失可以衡量图像在优化过程中与内容图像的相似性,计算式为

$$L_{\text{con}} = \|\phi_l(I_{\text{render}}) - \phi_l(I_{\text{content}})\|_2 \quad (10)$$

式中, I_{content} 表示内容图像, L_{con} 表示内容损失。

曲率损失能抑制控制点之间处于同一条直线上,保证曲线的弯曲度,因为本文共采用4个点控制Bezier曲线,每3个点即可控制一段曲线,所以本文

将4个节点控制的Bezier曲线分为两段子曲线优化, p_0, p_1, p_2 作为第1段子曲线的起点、控制点和终点; p_1, p_2, p_3 作为第2段子曲线的起点、控制点和终点。曲率计算式为

$$L_{se} = \frac{(s - c) + (e - c)}{s - e} \tag{11}$$

式中, s 表示曲线的起点, c 表示曲线的控制点, e 表示曲线的终点, L_{se} 表示曲线的曲率。根据式(11), 分别计算两段子曲线的曲率, 再对两段子曲线的曲率求均值。

两段曲线的总损失 L_{cur} 为

$$L_{cur} = \sqrt{L_{se1} + L_{se2}} \tag{12}$$

式中, L_{se1} 表示第1段曲线的曲率, L_{se2} 表示第2段曲线的曲率。

3 实验结果与分析

3.1 欺骗率

欺骗率表示虚假风格图像能够欺骗神经网络, 使神经网络判断虚假风格图像为真实风格图像的概率。为了客观评价风格化图像的风格特征与风格图像的风格特征是否接近, 本文将 Sanakoyue 等人 (2018) 提出的欺骗率作为客观评价指标。Sanakoyue 等人 (2018) 的方法将 wikiart 数据集 (Nichol, 2016) 作为分类训练集, 训练 VGG16 (Visual Geometry Group16-layer) 网络为 600+ 种艺术风格进行分类, 然后从 wikiart 数据集中选取部分类型的艺术图像作为风格图像, 将内容图像与风格图像进行风格迁移, 从风格图像中裁剪出部分子图像, 采用训练好的 VGG16 为子图像进行分类, 计算每种风格分类结果的平均值, 即平均欺骗率。如表 1 所示, VGG16 网络对 wikiart 测试集中 20 类真实艺术家图像的分类准确率为 0.652, 对内容图像的分类准确率为 0.02。将 AdaIN (Huang 和 Belongie, 2017)、WCT (whitening and coloring transforms) (Li 等, 2017)、Gatys 等人 (2016)、AST (arbitrary style transfer) (Sanakoyue 等, 2018)、Kotovenko 等人 (2021) 和本文方法采用平均欺骗率, 判断相同内容图像和风格图像下, 不同模型生成的风格迁移结果与真实艺术图像的风格相似度, 结果如表 1 所示。

然而, 欺骗率的分数高低并不能代表生成图像质量的好坏, 只能判断生成的风格图像与给定的风

格图像是否接近。因此, 增加了测试者欺骗率指标, 对神经网络生成的风格化图像进行主观评价, 由测试者根据自己的主观感受为结果图像打分。本次实验的测试者共有 30 人, 采用 20 幅风格图像与 20 幅内容图像交叉进行风格迁移, 生成 400 幅风格化图像, 由测试者将本文方法与 Kotovenko 等人 (2021)、Gatys 等人 (2016)、AST (Sanakoyue 等, 2018)、AdaIN (Huang 和 Belongie, 2017) 和 WCT (Li 等, 2017) 等 5 种方法风格迁移结果进行比较。欺骗率指标的分数越高表示风格化图像与真实艺术图像越相似, 表 1 展示了每种方法的欺骗率分数。本文方法在两个指标上获得的分数均为最高, 说明本文方法风格化图像在视觉效果上优于其他 5 种方法。

表 1 欺骗率

Table 1 Deception rate

方法	平均欺骗率	测试者 平均欺骗率
内容图像 (Lin 等, 2014)	0.002	-
wikiart 测试集 (Nichol, 2016)	0.652	-
Gatys 等人 (2016)	0.392	0.129
AST (Sanakoyue 等, 2018)	0.447	0.153
AdaIN (Huang 和 Belongie, 2017)	0.084	0.031
WCT (Li 等, 2017)	0.165	0.088
Kotovenko 等人 (2021)	0.542	0.271
本文	0.585	0.286

注: 加粗字体表示各列最优结果, “-” 表示非风格化图像无法计算测试者欺骗率数值。

3.2 实验时间比较

为了表明本文方法的高效性, 将本文方法与 GANILLA (Hicsonmez 等, 2020)、Paint Transformer (Liu 等, 2021) 和 StrokeNET (Zheng 等, 2023) 模型的运行时间进行比较。其中, GANILLA 为风格迁移模型, Paint Transformer 和 StrokeNET 为图像绘画式重建模型, 各个模型的渲染时间和训练时间如表 2 所示, 所有模型在渲染时采用相同的内容图像和风格图像。与 GANILLA 和 StrokeNET 相比, 本文方法渲染时间长, 但是本文方法通过图像分割算法, 取点生成曲线笔触参数, 无需额外的数据集训练模型, 缩短了程序运行时间。此外, GANILLA 模型仅适

用于插图风格图像的迁移,而本文方法可以生成多风格图像。与 Paint Transformer 和 StrokeNET 相比,本文方法生成的结果图像可以保留更多的图像细节。

表 2 实验时间
Table 2 Experimental time

方法	渲染/s	训练/h	数据集
GANILLA(Hicsonmez 等,2020)	0.39	3.82	需要
Paint Transformer(Liu 等,2021)	0.42	4.16	不需要
StrokeNET(Zheng 等,2023)	0.41	40	需要
本文	374	-	不需要

注:“-”表示该方法无需训练。

3.3 实验结果

图 6 展示了本文方法与 Wang 等人(2021)、Gatys 等人(2016)和 Kotovenko 等人(2021)等 3 种风格迁移算法生成的结果图像。与 Wang 等人(2016)的方法相比较,Wang 等人(2016)的结果图像更接近风格图像,但是 Wang 等人(2016)在图像上色阶段加入了噪声扰动,导致结果图像中生成一些比较杂乱的纹理,影响结果图像的整体视觉效果。本文方法虽然采用多尺度笔触渲染图像前景和背景,但

是结果图像整体减少了笔触杂乱堆积,而且笔触痕迹清晰,色彩艳丽,结果图像更接近艺术家的手绘作品。与 Gatys 等人(2016)的方法相比较,Gatys 等人(2016)的方法保留了较多内容图像的风格,如图 6(d)第 1 行,Gatys 等人(2016)的方法过多地保留了大象原生的皮肤色彩,并且图像中颜色分布不均匀,图 6(d)第 2 行,Gatys 等人(2016)生成的图像整体色彩与风格图像整体色彩不一致,本文方法生成的结果图像在保留内容图像结构的同时,图像风格更加接近风格图像,因为本文方法可以更好地提取风格图像的色彩信息,将色彩信息与内容图像融合,结果图像中体现出更多风格图像的色彩。与 Kotovenko 等人(2021)的方法相比,Kotovenko 等人(2021)生成的结果图像中,图像前景与背景基本完全融合,没有清晰的边界线,并且结果图像中的部分色彩饱和度高于风格图像中的色彩饱和度,图像整体色彩不协调,而本文方法生成的笔触纹理更加均匀、细腻、显著,生成图像的色彩更接近风格图像色彩,如图 6(f)第 2、4 行所示,本文方法生成的图像背景中灰色部分更少,图像饱满,前景与背景的边界更清晰,图像背景部分的笔触更多,结构保存完整。

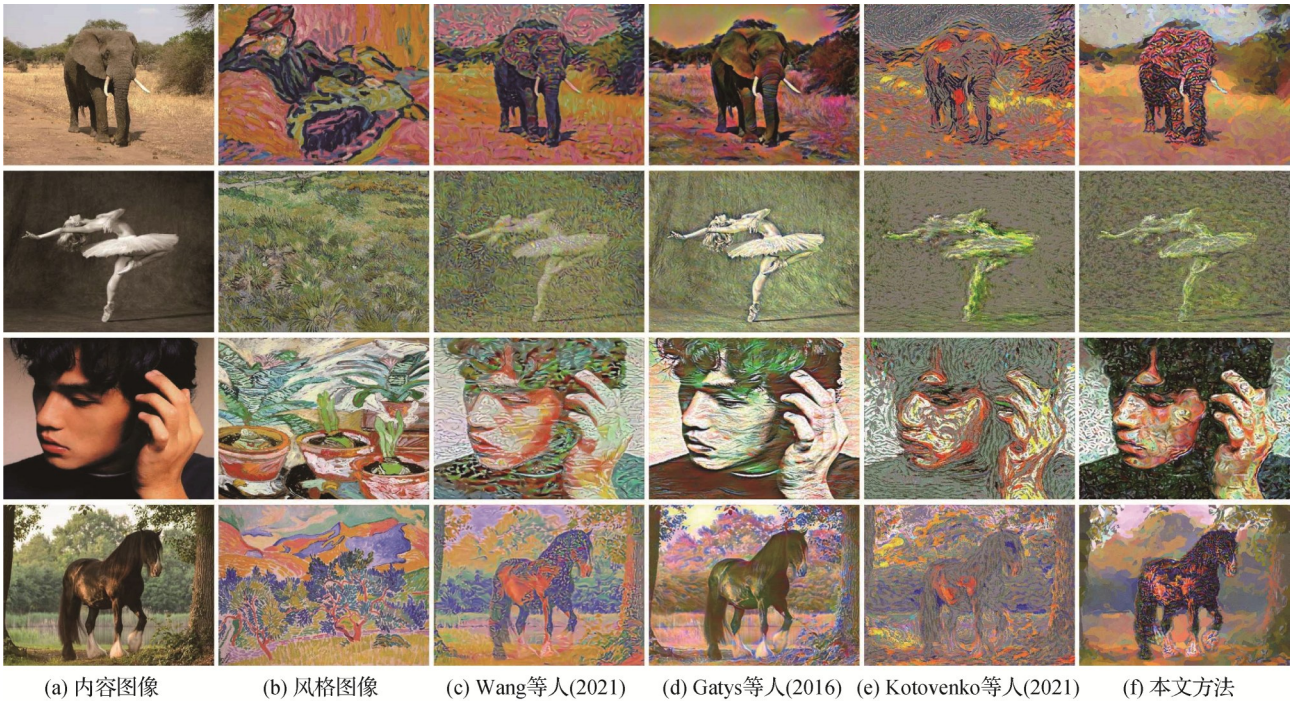


图 6 风格迁移效果图

Fig. 6 Style transfer results

((a) content images; (b) style images; (c) Wang et al. (2021); (d) Gatys et al. (2016); (e) Kotovenko et al. (2021); (f) ours)

3.4 图像重建

为了进行图像重建的对比,将本文方法与 Zou 等人(2021)和 Liu 等人(2021)的方法进行比较。所有模型均采用 python 和 pytorch 深度学习框架实现,在 Nvidia RTX 2080 Ti GPU 上训练,Zou 等人(2021)的方法需要自定义所生成的笔触数量,在图像分割与笔触渲染的过程中,计算资源的需求量逐渐增大,所以实验中尽可能地生成最多笔触。实验结果如图 7 所示。Zou 等人(2021)需要对内容图像分割后渲染,内容图像分割的数量决定渲染图像的细节数量,随着分割数量的增加,程序运行时间延长,消耗的资源也逐渐增加。Liu 等人(2021)的方法采用矩形笔触渲染图像,矩形笔触无法描绘出图像细节,而本文方法采用曲线笔触渲染图像,不仅能够在细节层面

上较好地还原源图像的细小纹理,而且运行时间短,如图 7 第 3 行中飞盘上的花纹、背景中砖块之间的缝隙、第 4 行中羊的眼睛、羊毛上的竖条纹理,本文方法保留的纹理比 Zou 等人(2021)和 Liu 等人(2021)的方法更多。此外,图 7 第 4 行,源图像中有 3 只羊,但是在 Zou 等人(2021)和 Liu 等人(2021)的方法中仅重建出两只羊,本文方法重建的图像中更好地保存了第 3 只羊的轮廓。与 Liu 等人(2021)的方法相比,Liu 等人(2021)的方法每次生成 8 个笔触,在生成 4 个大笔触的基础上,再叠加 4 个小笔触,对于纹理较为丰富的图像,为了保留图像细节,很容易产生大量小笔触叠加的效果,导致图像模糊,本文方法在分割后不相交的子图像中生成笔触,所以本文方法生成的图像在保留更多细节的前提下,不会产生大量小笔触堆叠的效果。

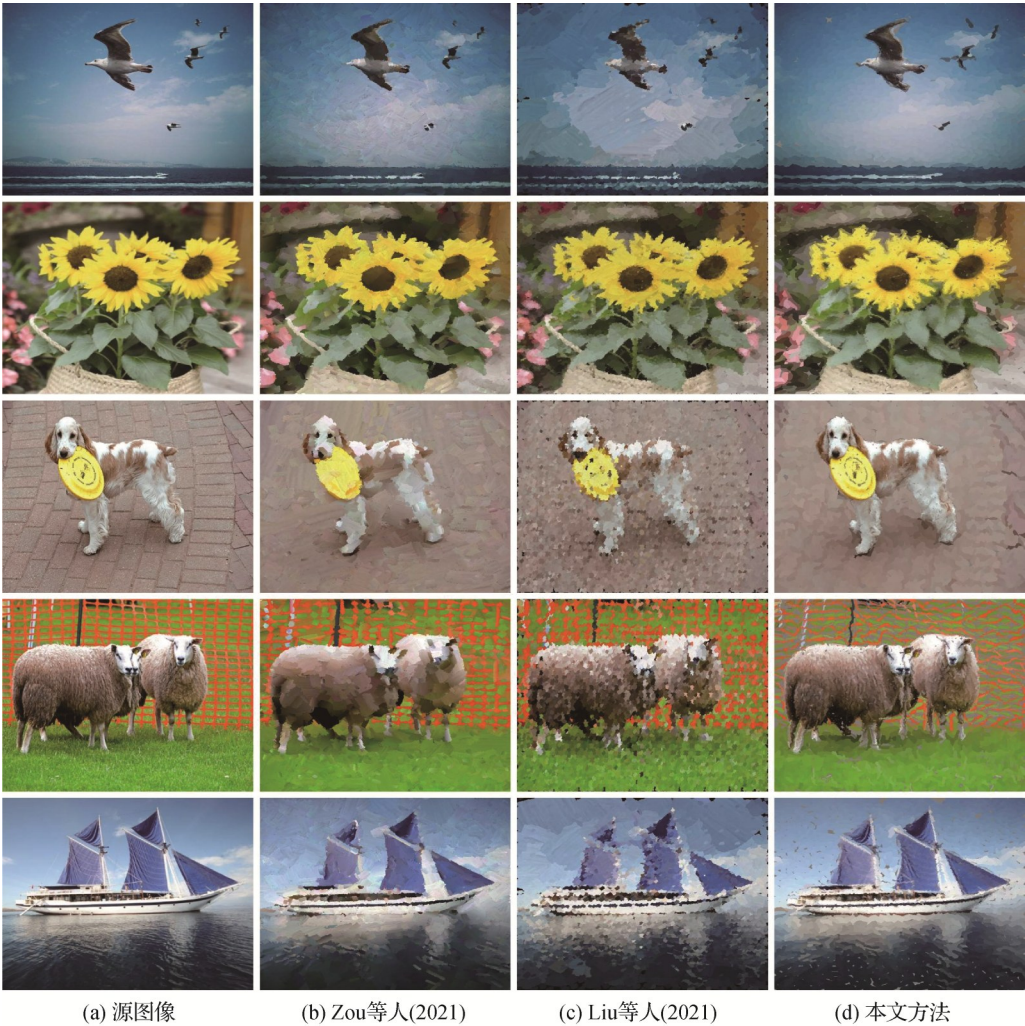


图 7 图像重建结果

Fig. 7 Image to painting results ((a) source images; (b) Zou et al. (2021); (c) Liu et al. (2021); (d) ours)

3.5 消融实验

笔触宽度是决定生成图像质量的重要因素之一,图像分割数量即生成笔触的数量,图像分割数量多,生成的笔触较小,图像分割数量少,生成的笔触较大。无论分割后子区域的大小,都需要生成的笔触能够覆盖整个子区域,减少图像中的空白部分。图 8 展示了 Kotovenko 等人(2021)方法生成的两幅图像,每幅图像均分割为 5 000 个子区域,生成 5 000 条笔触。可以看出,图像中仍然有大量灰色背景没有被笔触覆盖。

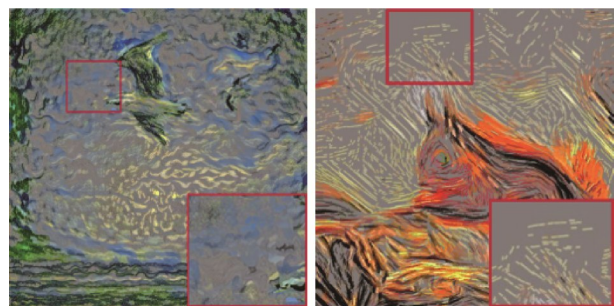


图 8 Kotovenko 等人(2021)方法的结果
Fig. 8 The result of Kotovenko et al. (2021) method

为了使生成的笔触能够填满画布,本文采用临界值确定细粒度笔触宽度和粗粒度笔触宽度。首先确定细粒度笔触的宽度临界值,保证细粒度笔触能够绘制出更多的图像细节,再确定粗粒度笔触的宽度临界值。

如图 9 所示,将细粒度笔触的临界值设置为 0.1 时,图像重建的均方误差(mean square error, MSE)损失最小。

确定细粒度笔触临界值后,对粗粒度笔触宽度临界值进行 MSE 损失的消融。如图 10 所示,当选择

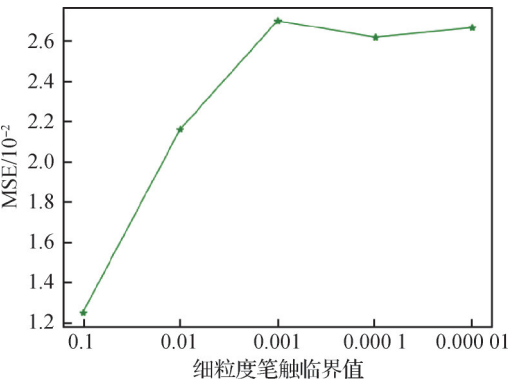


图 9 细粒度笔触 MSE 损失
Fig. 9 The MSE loss of thin strokes

$10^{-1} \times 100$ 作为粗粒度笔触和细粒度笔触的宽度临界值时,图像重建的 MSE 损失最低,并且图像中的笔触渲染更充分。

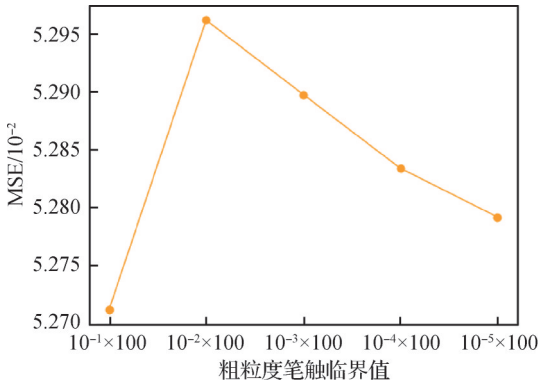
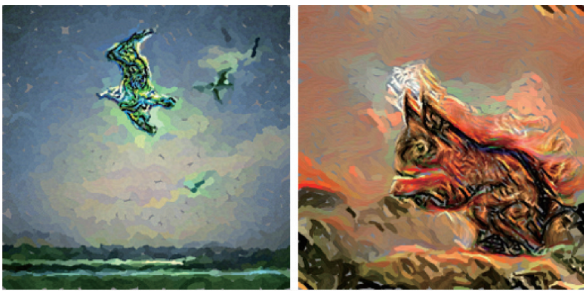


图 10 粗粒度笔触 MSE 损失
Fig. 10 The MSE loss of thick strokes

本文方法生成结果如图 11 所示。图 11(a)中的海鸥由 2 500 条笔触组成,图 11(b)中的松鼠由 3 000 条笔触组成。图 8 中 Kotovenko 等人(2021)生成的图像中有许多短小的直线笔触,图 8 右图中整幅图像更倾向于由直线笔触构成,无法凸显出笔触的曲线形态,因为 Kotovenko 等人(2021)的方法在优化过程中笔触颜色会趋近于背景颜色,导致图像看上去笔触较少,而且 Kotovenko 等人(2021)对图像整体分割数量多,所以生成的笔触短小,趋近直线。但若图像分割数量减少,图像整体呈现的笔触也会减少。本文方法生成的图 11 两幅图像的分割数量更少,生成的两幅图像在视觉效果上保留了更多的笔触,图像中的背景颜色更少,可以明显看到曲线笔触的形态。

图 12 展示了更多本文方法生成的结果图像。



(a) 2 500 条笔触 (b) 3 000 条笔触

图 11 本文方法生成结果

Fig. 11 Our results ((a) 2 500 strokes; (b) 3 000 strokes)



图 12 粗粒度笔触与细粒度笔触相结合的视觉效果
Fig. 12 The visual effect of the combination of thick and thin strokes ((a) content images; (b) style images; (c) style transfer results)

4 结 论

本文提出一种基于曲线笔触渲染的风格迁移算法,根据自定义掩膜区域内的超像素数量,对图像前景和背景分别生成不同尺度的曲线笔触,曲线笔触经过多次迭代优化,逐步优化笔触的位置、颜色和 control 点等参数,实现多风格迁移效果。实验结果表明,采用本文方法生成的结果图像色彩鲜艳艳丽,图像细小纹理保存更完整,并且生成的笔触更加贴近真实艺术作品的曲线笔触形状。由于 maskSLIC 算法分割后的图像块形状较为均匀,使得 Bezier 方程生成的曲线笔触不一定都在图像块内,从而导致图像中产生部分空白。未来将采用图像插值算法改善算法缺陷,并且当图像整体分割数量较多时,maskSLIC 算法的效率还有待进一步提高。此外,风格迁移网络仅能传输风格图像的色彩信息,下一步将考虑设计轻量级风格迁移算法,更好地提取风格图像的色彩特征和纹理特征,并采用更加客观、公正的指标进一步体现本文算法的有效性。

参考文献(References)

- Achanta R, Shaji A, Smith K, Lucchi A, Fua P and Süsstrunk S. 2012. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11): 2274-2282 [DOI: 10.1109/TPAMI.2012.120]
- Amami A, Azouz Z B and Alouane M T H. 2019. AdaSLIC: adaptive supervoxel generation for volumetric medical images. *Multimedia Tools and Applications*, 78 (3) : 3723-3745 [DOI: 10.1007/s11042-017-5563-3]
- An J, Huang S Y, Song Y B, Dou D J, Liu W and Luo J B. 2021. Art-Flow: unbiased image style transfer via reversible neural flows//*Proceedings of 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Nashville, USA: IEEE: 862-871 [DOI: 10.1109/CVPR46437.2021.00092]
- Ganin Y, Kulkarni T, Babuschin I, Ali Eslami S M and Vinyals O. 2018. Synthesizing programs for images using reinforced adversarial learning [EB/OL]. [2022-12-22]. <https://arxiv.org/pdf/1804.01118.pdf>
- Gatys L A, Ecker A S and Bethge M. 2015. A neural algorithm of artistic style [EB/OL]. [2015-08-26]. <https://arxiv.org/pdf/1508.06576.pdf>
- Gatys L A, Ecker A S and Bethge M. 2016. Image style transfer using convolutional neural networks//*Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition*. Las Vegas, USA: IEEE: 2414-2423 [DOI: 10.1109/CVPR.2016.265]
- Ghariba B, Shehata M S and McGuire P. 2022. Salient object detection using semantic segmentation technique. *International Journal of Computational Vision and Robotics*, 12(1): 17-38 [DOI: 10.1504/IJCVR.2022.119240]
- Hertzmann A. 1998. Painterly rendering with curved brush strokes of multiple sizes//*Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*. [s.l.]: ACM: 453-460 [DOI: 10.1145/280814.280951]
- Hertzmann A. 2003. A survey of stroke-based rendering. *IEEE Computer Graphics and Applications*, 23 (4) : 70-81 [DOI: 10.1109/MCG.2003.1210867]
- Hertzmann A, Jacobs C E, Oliver N, Curless B and Salesin D H. 2001. Image analogies//*Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*. [s.l.]: ACM: 327-340 [DOI: 10.1145/383259.383295]
- Hicsonmez S, Samet N, Akbas E and Duygulu P. 2020. Ganilla: generative adversarial networks for image to illustration translation. *Image and vision computing*, 95: #103886 [DOI: 10.1016/j.imavis.2020.103886]
- Huang X and Belongie S. 2017. Arbitrary style transfer in real-time with adaptive instance normalization//*Proceedings of 2017 IEEE International Conference on Computer Vision*. Venice, Italy: IEEE: 1510-1519 [DOI: 10.1109/ICCV.2017.167]
- Huang Z W, Zhou S C and Heng W. 2019. Learning to paint with model-based deep reinforcement learning//*Proceedings of 2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. Seoul, Korea (South) : IEEE: 8708-8717 [DOI: 10.1109/ICCV.2019.00880]
- Irving B. 2016. maskSLIC: regional superpixel generation with application to local pathology characterisation in medical images [EB/OL]. [2022-12-22]. <https://arxiv.org/pdf/1606.09518.pdf>

- Johnson J, Alahi A and Li F F. 2016. Perceptual losses for real-time style transfer and super-resolution//Proceedings of the 14th European Conference on Computer Vision (ECCV). Amsterdam, the Netherlands: Springer: 694-711 [DOI: 10.1007/978-3-319-46475-6_43]
- Kotovenko D, Wright M, Heimbrecht A and Ommer B. 2021. Rethinking style transfer: from pixels to parameterized brushstrokes//Proceedings of 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Nashville, USA: IEEE: 12191-12200 [DOI: 10.1109/CVPR46437.2021.01202]
- Li Y J, Fang C, Yang J M, Wang Z W, Lu X and Yang M H. 2017. Universal style transfer via feature transforms [EB/OL]. [2022-12-22]. <https://arxiv.org/pdf/1705.08086.pdf>
- Lin T W, Ma Z Q, Li F, He D L, Li X, Ding E R, Wang N N, Li J and Gao X B. 2021. Drafting and revision: Laplacian pyramid network for fast high-quality artistic style transfer//Proceedings of 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Nashville, USA: IEEE: 5137-5146 [DOI: 10.1109/CVPR46437.2021.00510]
- Lin T Y, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollár P and Zitnick C I. 2014. Microsoft coco: common objects in context//Proceedings of 2014 European Conference on Computer Vision. Zurich, Switzerland: 740-755 [DOI: 10.1007/978-3-319-10602-1_48]
- Liu S H, Lin T W, He D L, Li F, Deng R F, Li X, Ding E R and Wang H. 2021. Paint transformer: feed forward neural painting with stroke prediction//Proceedings of 2021 IEEE/CVF International Conference on Computer Vision. Montreal, Canada: IEEE: 6578-6587 [DOI: 10.1109/ICCV48922.2021.00653]
- Liu Z L, Zhu W and Yuan Z Y. 2019. Image instance style transfer combined with fully convolutional network and cycleGAN. Journal of Image and Graphics, 24(8): 1283-1291 (刘哲良, 朱玮, 袁梓洋. 2019. 结合全卷积网络与CycleGAN的图像实例风格迁移. 中国图象图形学报, 24(8): 1283-1291) [DOI: 10.11834/jig.180624]
- Nichol K. 2016. Painter by numbers, wikiart [EB/OL]. [2022-12-22]. <https://www.kaggle.com/c/painter-by-numbers>.
- Sanakoyue A, Kotovenko D, Lang S and Ommer B. 2018. A style-aware content loss for real-time HD style transfer//Proceedings of the 15th European Conference on Computer Vision. Munich, Germany: IEEE: 715-731 [DOI: 10.1007/978-3-030-01237-3_43]
- Song Y Z, Rosin P L, Hall P M and Collomosse J. 2008. Arty shapes//The 4th International Symposium on Computational Aesthetics in Graphics, Visualization, and Imaging. Lisbon, Portugal: 65-72 [DOI: 10.2312/compaesth/compaesth08/065-072]
- Wang P, Li Y J and Vasconcelos N. 2021. Rethinking and improving the robustness of image style transfer//Proceedings of 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Nashville, USA: IEEE: 124-133 [DOI: 10.1109/CVPR46437.2021.00019]
- Wang Z Z, Zhao L, Chen H B, Qiu L H, Mo Q H, Lin S H, Xing W and Lu D M. 2020. Diversified arbitrary style transfer via deep feature perturbation//Proceedings of 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Seattle, USA: IEEE: 7786-7795 [DOI: 10.1109/CVPR42600.2020.00781]
- Xie B, Wang N and Fan Y W. 2020. Correlation alignment total variation model and algorithm for style transfer. Journal of Image and Graphics, 25(2): 241-254 (谢斌, 汪宁, 范有伟. 2020. 相关对齐的总变分风格迁移新模型. 中国图象图形学报, 25(2): 241-254) [DOI: 10.11834/jig.190199]
- Xie N, Hachiya H and Sugiyama M. 2013. Artist agent: a reinforcement learning approach to automatic stroke generation in oriental ink painting. IEICE Transactions on Information and Systems, E96.D(5): 1134-1144 [DOI: 10.1587/transinf.E96.D.1134]
- Zhang Y L, Fang C, Wang Y L, Wang Z W, Lin Z, Fu Y and Yang J M. 2019. Multimodal style transfer via graph cuts//Proceedings of 2019 IEEE/CVF International Conference on Computer Vision. Seoul, Korea (South): IEEE: 5942-5950 [DOI: 10.1109/ICCV.2019.00604]
- Zheng N Y, Jiang Y F and Huang D J. 2023. StrokeNet: a neural painting environment [EB/OL]. [2022-12-22]. <https://openreview.net/forum?id=HJxwDiActX>
- Zou Z X, Shi T Y, Qiui S, Yuan Y and Shi Z W. 2021. Stylized neural painting//Proceedings of 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Nashville, USA: IEEE: 15684-15693 [DOI: 10.1109/CVPR46437.2021.01543]

作者简介

饶师瑾,女,硕士研究生,主要研究方向为计算机视觉和风格迁移算法。E-mail: raoshijin2020@mail.ynu.edu.cn

钱文华,通信作者,男,教授,博士生导师,主要研究方向为图像处理和计算机视觉。E-mail: whqian@ynu.edu.cn

张结宝,男,硕士研究生,主要研究方向为人工智能安全、凸优化、信息论和神经网络的可解释性。

E-mail: zhangjiebao2014@mail.ynu.edu.cn