E-mail: jig@aircas.ac.cn
Website: www.cjig.cn
Tel: 010-58887035
中国图象图形学报
JOURNAL OF IMAGE AND GRAPHICS

中图法分类号:TP18;TP391.4　　**文献标识码**:A　　**文章编号**:1006-8961(2026)03-0840-10

论文引用格式:Zheng S Y, Chen J Z, Zhu X D and Li K Y. 2026. Vector graphics style transfer guided by cross-modal style evaluation models. Journal of Image and Graphics, 31(3):0840-0849(郑舒洋, 陈佳舟, 朱欣定, 李凯勇. 2026. 跨模态风格评分模型引导的矢量图风格迁移. 中国图象图形学报, 31(3):0840-0849)[DOI:10.11834/jig.250310]

跨模态风格评分模型引导的矢量图风格迁移

郑舒洋[1],陈佳舟[1*],朱欣定[1],李凯勇[2]

1. 浙江工业大学计算机科学技术学院,杭州　310023; 2. 青海民族大学智能科学与工程学院,西宁　810007

摘　要:**目的**　矢量图风格迁移旨在将特定艺术风格应用于目标内容,同时保持其结构。然而,现有方法仅提取特定图像特征作为风格,导致迁移结果与人类视觉感知存在差距。为解决这一问题,提出了一种跨模态风格评分模型引导的矢量图风格迁移方法(vector graphics style transfer guided by cross-modal style evaluation models,CLIPVGStyler),旨在利用CLIP(contrastive language-image pre-training)模型的跨模态理解能力实现更符合感知的矢量图风格迁移。**方法**　首先,将风格描述文本或风格参考图像作为输入,通过预训练的CLIP模型将其编码至共享语义信息的潜在空间,无需额外训练;然后,计算风格文本/图像编码与当前画面编码之间的余弦距离,构建风格损失,而内容损失则由像素域的L2距离和CLIP编码后的余弦距离共同组成;最后,通过迭代优化画面参数(路径控制点、颜色等)最小化联合损失函数(风格损失+内容损失),最终生成符合目标风格的矢量图。**结果**　与先进的3种方法进行了风格迁移效率和风格质量比较,CLIP评分和用户调研的评分均更高,CLIP评分相比排名第2的模型高了0.003,用户调研的得分相比排名第2的模型高了0.36分,迭代速度排第2。实验结果表明,相比现有矢量图风格迁移方法,该方法生成的图像细节更精细,风格迁移效果更显著且符合人类视觉感知。此外,该方法在迭代优化速度上也展现出优势。**结论**　本文提出的基于CLIP跨模态风格评分的CLIPVGStyler方法,有效融合了文本和图像的语义信息,成功解决了现有方法风格迁移结果与人类视觉感知不符的问题,显著提升了矢量图风格迁移的效果和效率。

关键词:矢量图;风格迁移;跨模态;少样本;CLIP

Vector graphics style transfer guided by cross-modal style evaluation models

Zheng Shuyang[1], Chen Jiazhou[1*], Zhu Xinding[1], Li Kaiyong[2]

1. College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou　310023, China;

2. School of Intelligence Science and Engineering, Qinghai Minzu University, Xining　810007, China

Abstract: **Objective** Style transfer is a popular research problem in computer graphics, aimed at applying one style to another content to give it a new visual style. Since the introduction of the Gram matrix-based method for representing style, many deep learning-based methods have emerged in the field of style transfer, which have achieved good results through convolutional neural networks. However, these studies are limited to raster images and difficult to directly apply to style transfer in vector images mainly because the basic elements of vector images are different from bitmap images and cannot be convolved like pixels. Iteratively updating network parameters through reverse traditional propagation is also impossible. Vector graphics describe images using geometric shapes, which have the advantages of scale invariance, small data volume, and flexible editing. They are widely used in fields such as illustration and web design. Therefore, the style transfer

收稿日期:2025-07-16;**修回日期**:2025-09-04;**预印本日期**:2025-09-11

*通信作者:陈佳舟　cjz@zjut.edu.cn

基金项目:国家自然科学基金项目(62172367)

Supported by: National Natural Science Foundation of China(62172367)

of vector graphics is also one of the research hotspots in computer graphics. The existing methods face serious limitations: they typically only extract specific low-level features (such as strokes or textures) from reference style images and cannot capture the overall and multifaceted visual perception of style by humans (including color, texture, strokes, and thematic elements). This can lead to inconsistent perceptual results, such as unnatural color shifts or distorted content. The dependence on style images also limits control and expressiveness. **Method** CLIPVGStyler proposes a contrastive language-Image pre-training (CLIP)-based cross-modal style scoring model to supervise vector graphics parameter optimization. The core innovation lies in using CLIP's zero-shot style discrimination capability as a perceptual loss function. The method takes an input vector graphic defining the content and a textual style description. Optionally, a small cache (4～16 images) representing a novel style (StyleCache) can be provided for few-shot style adaptation. The differentiable rasterizer called DiffVG is used to render the current vector graphic state during optimization. The loss function consists of two key components. The style loss measures the dissimilarity between the current rendered image and the target style. It primarily uses the cosine distance between the CLIP encodings of current vector graphic and style. For few-shot adaptation, an additional loss term ($1-S$_cache) is added, where S_cache is the average cosine similarity between the current vector graphic and the CLIP encodings of the StyleCache images. The content loss ensures content preservation and is a weighted combination ($\alpha = 0.25$) of the pixel-level L2 distance between current vector graphic and content image and the cosine distance between their CLIP encodings. Crucially, the rendered current vector graphic undergoes robust image augmentation (multi-scale random cropping with white padding and random perspective transformations) before CLIP encoding to enhance geometric invariance and prevent overfitting. Negative text prompts ("bad image", "low quality", "blurry image") are also incorporated to improve output quality. The combined loss (L_style' + L_content) is minimized iteratively using gradient descent to update the vector graphic's path parameters (shapes, colors). An early stopping mechanism halts optimization if the loss plateaus for 10 consecutive iterations, typically converging within 50−150 steps. **Result** Comprehensive experiments demonstrate the effectiveness and efficiency of CLIPVGStyler. Quantitatively, using CLIP similarity scores as a semantic-alignment metric, CLIPVGStyler achieved the highest score (0.256) compared with baselines (StyleCLIPDraw: 0.241, VectorPainter: 0.253, VectorNST: 0.239) when evaluating the semantic fidelity of the output to the textual style prompt T_style. A large-scale user study (360 participants) employing a 10-point scale for blind A/B testing further confirmed the perceptual superiority of CLIPVGStyler. It received the highest average rating (7.5), significantly outperforming StyleCLIPDraw (3.2) and VectorNST (3.86), and slightly exceeding the well-regarded but inefficient VectorPainter (7.14). Qualitatively, CLIPVGStyler generated results with more pronounced and holistic style transfer across diverse styles (e.g., watercolor, *One Piece* anime, ink wash, sketch), successfully capturing complex stylistic elements such as color palettes and thematic coherence without the distortions prevalent in other methods (e.g., unnatural color bleeding in VectorPainter, excessive contour distortion in VectorNST, or chaotic composition in StyleCLIPDraw). Crucially, it maintained superior content integrity compared with alternatives when using text-based style control. The proposed few-shot style adaptation mechanism effectively enhanced the transfer of novel or rare styles unseen by CLIP during pre-training. Ablation studies confirmed the critical role of the primary style loss based on text (S_Text), which alone handled common styles effectively. The S_Cache component proved essential for boosting novel style perception and provided marginal refinement for common styles but was ineffective when used alone. While S_Text could operate independently, S_Cache required the foundation of S_Text for optimal results. Performance-wise, CLIPVGStyler (avg. 19 min 10 s) demonstrated a substantial speed advantage over the highly effective but impractical VectorPainter (842 min 32 s) and was faster than VectorNST (24 min 14 s), approaching the speed of the lower-quality StyleCLIPDraw (14 min 27 s). **Conclusion** CLIPVGStyler presents a groundbreaking approach to vector graphics style transfer by introducing a CLIP-based cross-modal style scoring model. This model leverages CLIP's inherent human-aligned style recognition capabilities to supervise the optimization of vector graphic parameters, leading to results that are significantly more perceptually coherent than prior work. Key contributions include the novel formulation of style loss using CLIP encodings of textual style descriptions (T_style), enabling precise, high-level control over the desired artistic expression. The method introduces an efficient few-shot adaptation mechanism (StyleCache) that allows the system to handle novel or niche styles without any additional training, compensating for CLIP's limitations on unseen data. Extensive experimental validation confirms that CLIPVGStyler generates vector

graphics with superior style transfer quality, achieving higher semantic alignment (CLIP score) and user preference ratings than state-of-the-art methods such as StyleCLIPDraw, VectorPainter, and VectorNST. It accomplishes this task with high computational efficiency, particularly compared with the high-quality but prohibitively slow VectorPainter. Limitations include potential minor artifacts upon close inspection and the possibility of inadvertently transferring non-style features. Overall, CLIPVGStyler establishes a new paradigm for perceptually guided, efficient, and controllable vector graphics style transfer.

Key words: vector graphics; style transfer; cross-modal; few-shot; contrastive language-image pre-training(CLIP)

0 引 言

早期风格迁移的研究主要集中在栅格图上。Gatys等人(2016)发现,利用预训练的卷积神经网络可以提取图像中的风格特征,并通过计算内容损失和风格损失,使用梯度下降法优化输入图像的像素,可以生成具有指定风格的图像。然而,这种方法的生成效率较低。此后几年,许多研究者提出各种方法以提升风格迁移的质量和效率,其中一系列工作专注于正则化特征。Ulyanov等人(2016)提出的方法对一种风格图像训练后即可实现实时风格迁移。Huang和Belongie(2017)提出了自适应的实例正则化,实现了任意风格的实时风格迁移。此外,近年来也有利用扩散模型实现风格迁移的研究(孙梅婷等,2023;张宇欣 等,2025),能够实现语义层面的风格迁移,但可能导致内容的意外变化。

然而,这些研究局限于栅格图像,难以直接应用于矢量图的风格迁移,主要原因是矢量图基本元素和位图不同,不能和像素一样卷积,无法直接通过反向传播来迭代更新网络参数。另一方面,矢量图用几何图形描述图像,具有缩放不变性、数据量小和编辑灵活等优点,在插画绘制、网页设计和数字娱乐等领域应用广泛。因此,矢量图的风格迁移也是计算机图形学的研究热点之一。

为了解决矢量图不可微的问题,DiffVG(differentiable vector graphics)(Li等,2020)提出了一个针对矢量图形的可微渲染器,它允许直接优化矢量图中的元素和路径,例如贝塞尔曲线而不是像素,后续基于神经网络生成矢量图形的研究均以此可微渲染器为基础。基于此,近年来逐步出现了针对矢量图的风格迁移方法,包括StyleCLIPDraw(Schaldenbrand等,2022),VectorPainter(Hu等,2025)和VectorNST(neural style transfer)(Efimova等,2023)。StyleCLIPDraw利用CLIP(contrastive language-image pre-training)(Radford等,2021)生成内容,应用VGG(Visual Geometry Group)提取的图像风格特征来迁移风格,由于CLIP倾向提供高层语义,缺乏画面精细的控制,并且没有利用图像直接控制画面,因此生成的画面整体较为杂乱。VectorNST则是用VGG提取图像特征,用轮廓约束内容不变,用LPIPS(learned perceptual image patch similarity)(Zhang等,2018)感知损失监督风格变化,直接对输入的矢量图进行迭代优化,由于只改变原有的图形参数,并且未引入额外形状,因此该方法的风格迁移结果误差往往较大。VectorPainter通过提取风格图像的笔触,内置扩散模型(stable diffusion)(Rombach等,2022)生成内容图像,再用提取的笔触配合LPIPS损失重新排列成内容图像的画面,能够生成质量很高的风格化图像,但是要求风格参考图像有较明显的笔触,否则迁移的质量会大大下降。由于优化步骤较为复杂和烦琐,VectorPainter算法效率较低,生成一个矢量图往往需要若干小时。

一幅绘画的风格包括多种因素,如主题、色彩、纹理和笔触等。然而,上述矢量图风格迁移方法只是提取了风格参考位图的特定特征,难以有效表达人类对风格的复杂视觉感知,也无法对想要的风格因素进行单独指定。以图1中VectorPainter(Hu等,2025)的风格迁移结果为例,该方法侧重风格笔触,但是在风格迁移时将色彩也绑定一起迁移了,导致部分颜色失真,如将左图花朵的红色也迁移到了右图椰子树的叶子和沙滩上,使椰子树叶和沙滩呈现为不合理的红色。

为了让风格迁移更符合人类视觉感知,本文的主要思路有两个方面。1)提出一种跨模态风格评分模型引导矢量图的生成(主要是参数的迭代更新)。考虑到CLIP由大量图像文本对进行训练,其风格识别能力与人类对风格的认知具有较高的一致性,因

图 1　VectorPainter 风格迁移示意图

Fig. 1　Illustration of VectorPainter style transfer

此用 CLIP 构建该风格评分模型。基于 CLIP 的矢量图参数迭代更新不再需要低效的扩散生成模型，避免了重要细节特征的丢失。2）提出用文本描述风格代替风格参考位图。在文本—图像跨模态模型中，视觉与语言共享同一潜在空间；表示风格的文本蕴含丰富信息，而这些信息无法简单通过卷积网络提取图像特征完全表达。预训练的跨模态大模型通过海量的风格图像与文本数据的训练，能够学习到更符合人类认知的特征，因此利用文本表示画面风格更为有效。

虽然大模型可以很好地表达常见的风格特征，但对于少见的或全新的风格感知不够，依然容易导致风格评分模型不准确。为此，本文的风格评分方法在文本描述的基础上提出 few-shot 的图像新风格感知：给定 4～16 幅同一风格不同内容的风格参考图像，在不加训练的情况下能否自动感知新风格特征，并实现矢量图的风格迁移。

综上所述，本文的贡献主要包括：1）首次提出了基于 CLIP 的跨模态风格评分模型，它能够引导矢量图的风格迁移，使风格迁移解决更符合人类的视觉感知。此外，用文本代替图像指定风格，更加准确和可控。2）首次提出了 few-shot 的 CLIP 新风格感知方法，补充 CLIP 大模型对未知风格的不足。只需输入 4～16 幅代表新风格的图像即可实现对 CLIP 未见的风格进行迁移，且无需训练，使用便捷。3）大量实验表明，本文提出的风格迁移方法整体流程简洁高效，迭代优化速度较快，能够生成比已有工作更好的风格迁移矢量图。

1　本文方法

1.1　算法框架

本文方法的整体框架如图 2 所示，使用 DiffVG（differentiable vector graphics rasterzation）作为可微渲染器，给定表示内容的矢量图和表示风格的文本提示，每次迭代渲染后的图像分别由内容保留模块和风格评分模块计算损失，并反向传播梯度更新图形的形状和颜色等参数，迭代到整体损失收敛时即得到最终的矢量图。当风格为小众风格或者新风格时，还可以给出 4～16 幅能够代表该风格的图像来构建风格特征，以提高新风格的矢量图风格迁移质量。

1.2　风格评分模块

风格评分模块由两部分组成，分别是文本对比和缓存图像对比，其中缓存图像对比为可选部分，风

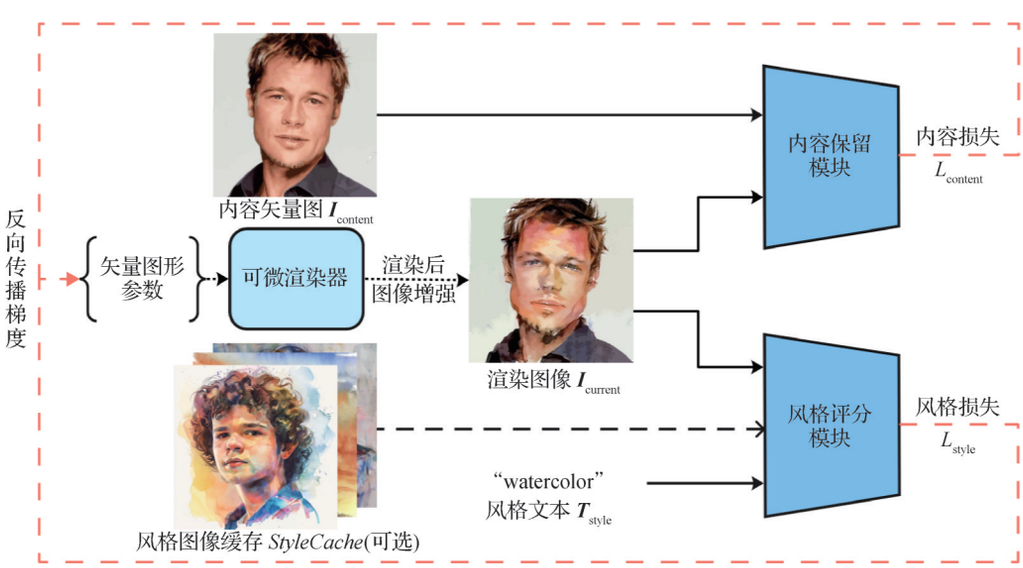

图 2　本文方法框架图

Fig. 2　The algorithm framework diagram of this article's method

格评分主要由文本对比来控制风格，风格图像缓存主要用于强化新风格感知能力。图像缓存将在下一节详细介绍。

文本对比首先计算增强后的图像$\boldsymbol{I}_{\text{currunt}}$和风格文本$\boldsymbol{T}_{\text{style}}$的风格损失。两者由CLIP编码后计算余弦相似度$S_{\text{Text}}$。具体为

$$S_{\text{Text}} = sim(\boldsymbol{I}_{\text{currunt}}, \boldsymbol{T}_{\text{style}}) \tag{1}$$

$$sim(\boldsymbol{A}, \boldsymbol{B}) = \frac{E(\boldsymbol{A}) \cdot E(\boldsymbol{B})}{\left\| E(\boldsymbol{A}) \right\| \cdot \left\| E(\boldsymbol{B}) \right\|} \tag{2}$$

式中，$E(\boldsymbol{A})$表示CLIP对$\boldsymbol{A}$的编码，$sim(\boldsymbol{A}, \boldsymbol{B})$表示两者CLIP编码的余弦相似度。由1减去后得到的余弦距离作为L_{style}。即

$$L_{\text{style}} = 1 - S_{\text{Text}} \tag{3}$$

L_{style}用于指导画面风格变化，由于CLIP具有识别图像风格的能力，因此可以作为辨别器监督画面变化。

1.3 Few-shot新风格感知

虽然上述基于CLIP大模型的风格迁移对于大部分风格都有效，但对于全新的或极为少见的风格，有时会出现迁移效果不好的问题。如果要专门进行风格识别任务，还需要额外训练。经过实验发现，线性探针和CLIP-Adapter（Gao等，2023）等微调方法对于矢量图风格识别任务并不适用。为此，提出一种few-shot的风格存储方法，使本文方法也能适应新风格或者小众风格。在输入4～16幅表示此类风格的不同图像后，能够获得识别此类风格特征的能力。具体而言，将存储的风格图像经过CLIP编码后保存，将每次迭代的$\boldsymbol{I}_{\text{currunt}}$分别与保存的特征向量计算余弦相似度，并取平均值得到S_{Cache}。具体为

$$S_{\text{Cache}} = Mean(sim(\boldsymbol{I}_{\text{currunt}}, StyleCache)) \tag{4}$$

式中，*StyleCache*为一幅风格图像缓存，*Mean*为求平均。其中相似的部分与风格特征较为接近，该相似度数值即可代表这些图像与目前图像$\boldsymbol{I}_{\text{currunt}}$的相似度。$S_{\text{Cache}}$结合$S_{\text{Text}}$可得到加强后的风格损失$L'_{\text{style}}$。即

$$L'_{\text{style}} = L_{\text{style}} + (1 - S_{\text{Cache}}) \tag{5}$$

1.4 内容保留模块

每次迭代更新后渲染得到的图像，需要预先经过图像增强处理。借鉴之前的相关工作，最终采用多尺度的随机裁剪结合透视变换的图像增强策略，首先通过带白色填充的随机裁剪提取局部特征，配合随机透视变换转换观察图像视角，最后使用小幅随机裁剪，得到增强后的图像$\boldsymbol{I}_{\text{currunt}}$。这些图像增强提升模型对几何变换的健壮性，同时保持核心内容完整性。此外，添加了反向文本提示“bad image”、“low quality”、“blurry image”以提升风格迁移画面质量。

内容保留模块计算内容图像$\boldsymbol{I}_{\text{content}}$和风格迁移后的图像$\boldsymbol{I}_{\text{currunt}}$的内容损失，由两部分组成。具体为

$$L_{\text{content}} = \alpha \cdot L_2(\boldsymbol{I}_{\text{currunt}}, \boldsymbol{I}_{\text{content}}) + (1 - \alpha) \cdot L_{\text{CLIP}} \tag{6}$$

式中，L_2是$\boldsymbol{I}_{\text{content}}$和$\boldsymbol{I}_{\text{currunt}}$的像素域的$L_2$距离，$\alpha$是两个损失的权重因子（根据不同取值试错，$\alpha = 0.25$时，整体实验结果表现较好，$\alpha$太小会导致画面扭曲，$\alpha$太大则导致迁移结果不明显，因此本文最终所有实验取$\alpha = 0.25$），$L_{\text{CLIP}}$是CLIP将$\boldsymbol{I}_{\text{content}}$和$\boldsymbol{I}_{\text{currunt}}$编码后对比两者特征向量的余弦距离。具体为

$$L_{\text{CLIP}} = 1 - sim(\boldsymbol{I}_{\text{currunt}}, \boldsymbol{I}_{\text{content}}) \tag{7}$$

L_2和L_{CLIP}这两项计算$\boldsymbol{I}_{\text{content}}$和$\boldsymbol{I}_{\text{currunt}}$两者的像素差异和CLIP编码差异并对比计算损失，抑制了由风格损失引起的画面过度变化，确保当前画面与内容图像保持一定相似性，以保证画面不会出现意外的变形。

2 实验结果与分析

2.1 实验设置

2.1.1 实验环境

本文实验在以下硬件环境下进行：显卡为RTX 4070（12 GB显存），CPU为13th Gen Intel（R）Core（TM）i5-13600KF（3.50 GHz），内存为32.0 GB RAM，操作系统为Ubuntu20.04，编程语言为python3.6，CUDA版本为11.3。

2.1.2 实现细节

CLIP使用的基础模型为ViT-B/32，本文方法迭代次数在50～150次之内。由于迭代达到一定步数时容易出现类似过拟合的画面幻觉现象，因此在实验中引入早停机制，若连续10次迭代损失无明显优化，则停止训练。

2.1.3 数据准备

Adobe illustrator是一款先进的商业矢量图形设计工具。本文利用其图像描摹功能将光栅图像转化为一组不重叠的填充曲线，以此实现矢量化处理。

矢量化精度可通过控制目标颜色的数量进行调节，目标颜色数量越多，矢量化精度越高。

2.1.4　评价指标

鉴于本文方法采用文本表示风格，且所生成的风格更侧重于全局语义层面，传统的对比评估分数并不适用。常用的 LPIPS 和 FID（Fréchet inception distance）指标主要用于对两个图像进行分析比较，而本文采用 CLIP 分数进行定量评估，同时结合主观的用户研究和定性评估的方式。

2.2　对比实验

本文将所提方法与现有的矢量图风格迁移方法进行对比分析，包括 StyleCLIPDraw、VectorNST 和 VectorPainter。这 3 种对比方法均采用图像表示风格，并且 StyleCLIPDraw 和 VectorPainter 使用文本表示内容。因此，如图 3 所示，在对比实验中，分别采用较为匹配的文本和图像输入表示风格和内容。

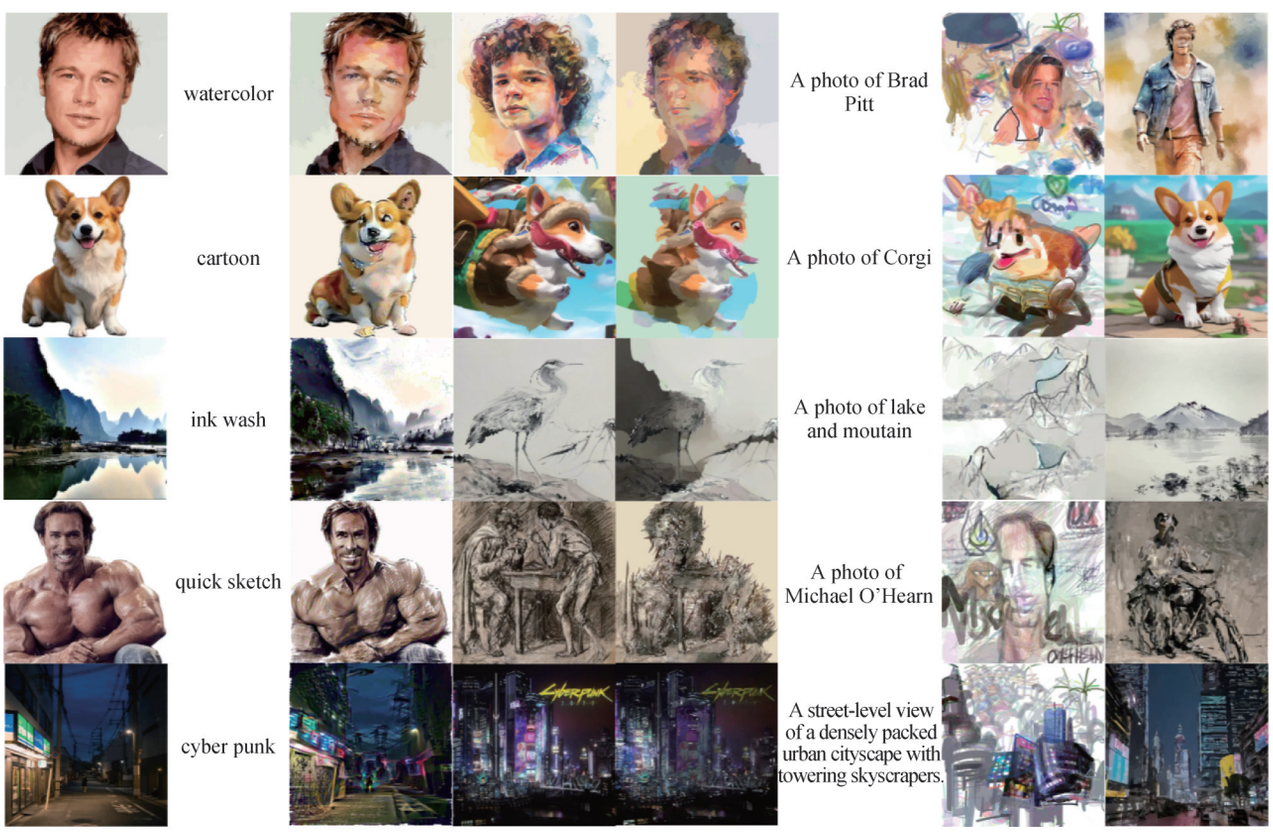

图 3　对比实验

Fig. 3　Comparative experiments ((a) content vector graphic inputs (ours, VectorNST); (b) style text inputs of our method; (c) results of our method; (d) style image inputs (VectorNST, StyleCLIPDraw, VectorPainter); (e) results of VectorNST; (f) content text inputs (StyleCLIPDraw, VectorPainter); (g) results of StyleCLIPDraw; (h) results of VectorPainter)

2.2.1　定量对比

通过 CLIP 指标对风格迁移质量进行评估，具体而言，以各方法输入风格与结果由 CLIP 编码的余弦相似度作为评估指标。该评分能够反映生成画面与目标风格的契合程度。由于本文方法与其他方法风格输入不同，风格图像不作为本文方法风格参考，因此对比风格图像相似度的数值不能反映本文方法的风格迁移效果。从表 1 可以看出，在数值上与风格文本的对比中本文相似度最高，因此能够证明本文方法结果在语义层面的风格表现较好。

2.2.2　用户研究

为评估本文方法风格迁移效果，邀请 360 名受

表1　各方法风格迁移结果在CLIP评分中的比较

Table 1　Comparison of style transfer results of various methods in CLIP score

方法	与风格文本相似度 ↑
VectorNST	0.239
StyleCLIPDraw	0.241
VectorPainter	0.253
本文	**0.256**

注：加粗字体表示最优结果，“↑”表示值越高越好。

试者参与盲测，采用问卷星进行A/B对比。每组任务呈现输入原图及4个匿名输出，用户通过10分制矩阵量表评估各结果（1：严重失真，10：完美还原）。最终统计各方法均分如表2所示，参与者对本文方法的结果评分最高。

表2　用户对各方法迁移结果评分平均值

Table 2　The average rating of users on the migration results of each method

方法	评分 ↑
StyleCLIPDraw	3.20
VectorNST	3.86
VectorPainter	7.14
本文	**7.50**

注：加粗字体表示最优结果，“↑”表示值越高越好。

2.2.3　定性对比

如图3所示，图3（e）中，VectorNST的风格迁移结果呈现出较为明显的轮廓迁移特征，较难看出原本的内容，这是由于其方法主要依赖于轮廓和LPIPS损失进行监督，导致其风格表现较差。图3（g）中，StyleCLIPDraw直接套用位图的风格迁移技术，构图较为混乱，并且风格表现不明显。图3（h）中，VectorPainter由扩散模型生成风格化位图，再用特定的矢量图形排列拼凑，因此画面整体较好，但由于矢量图形在逼近扩散模型时画面质量必然有所下降，难以复刻精细笔触。并且扩散模型生成画面会出现不必要的内容，导致文本提示描述的内容占比较少，尤其是在对生成画面内容精度要求较高的情况下，其内容表现不佳。例如，在进行名人的风格迁移时，生成结果中的人脸难以辨认。对比图3（a）和图3（c），可以发现在保证内容的可辨识度的同时，本文方法可以较好地表现风格。相比其他方法，本文的风格比较明显且较全面，但根据第2行和第3行可知画面可能会出现空隙。

图4为本文部分实验结果展示，可见本文方法对不同矢量图不同风格都具有风格迁移能力。图4第2列的矢量图本身为水彩画，因此进行相近风格的风格迁移后画面变化小。同时，对小众风格（“one piece”）也有风格迁移能力。

2.2.4　性能对比

在实验中，本文方法与VectorNST均基于输入矢量图进行迭代优化。其中，表3中所提及的输入矢量图有9 643个路径，VectorNST迭代100次，而本文方法迭代81次。StyleCLIPDraw使用256个三次贝塞尔曲线，迭代次数达到500次。VectorPainter则采用5 000个三次贝塞尔曲线，在模仿笔触阶段迭代250次，在合成画面阶段迭代2 000次。各方法的性能表现如表3所示，可以看出，画面表现较差的StyleCLIPDraw的速度最快，VectorNST的速度慢于本文方法，而VectorPainter的速度最慢，缺乏实用性。

2.3　消融实验

本文方法采用早停机制（在2.1.2小节实现细节中介绍），迭代约在50～150次之间完成，由于风格明显程度与内容保留程度有一定的互斥，一些抽象风格种类（如anime、cartoon等）在迭代后期风格较明显的情况下可能会出现过拟合现象。如图5所示，最佳的风格迁移结果出现在第59次迭代，但在第99次迭代时画面已经出现预期外的变化。不同的图像对不同风格进行迁移达到最佳评分的迭代次数不同，因此本文采用的早停机制能及时地停止迭代，避免过拟合。

图6展示了本文的第1个消融实验，可以直观地看到各模块的影响。对比图6（c）和图6（d）可看出，没有内容损失控制时，风格表现非常明显，但其内容会出现较大变化。CLIP本身具有一定的风格识别能力，对比图6（c）和图6（e）可发现，没有S_{Cache}的作用，只应用S_{Text}时，模型已经具备较好的风格迁移能力。但会出现更多的空隙，并且左侧树木无法辨认，可见一定程度上能够抑制对内容的破坏，同时保持较明显的风格。对比图6（c）和图6（f）可发现，只用S_{Cache}会出现扭曲的笔触，并且少量图像缓存不足以提取足够的风格特征，只能作为辅助模块，单独

图 4　本文方法对不同矢量图进行不同风格迁移的实验结果

Fig. 4　Experimental results on the transfer of different styles of vector graphics using our method

表 3　各方法完成一次风格迁移耗时

Table 3　The time cost for each method to complete a style transfer once

方法	耗时
VectorPainter	842 min 32 s
VectorNST	24 min 14 s
本文	19 min 10 s
StyleCLIPDraw	**14** min **27** s

注：加粗字体表示最优结果。

应用效果较差。

针对 S_{Text} 和 S_{Cache} 进行了更多的消融实验，以表明 S_{Text} 和 S_{Cache} 的效果。如图 7 所示，通过 Michael O’Hearn 的肖像分别应用水彩、速写、海贼王（小众风格）和水墨画风格迁移，对风格评分模块中的 S_{Text} 和 S_{Cache} 的控制效果进行了对比。

此处结合图 7 对 S_{Text} 功能的必要性和单独处理风格迁移能力进行说明。对于水彩风格，仅 S_{Cache} 和仅 S_{Text} 的结果都表现出明显的风格，而同时使用 S_{Text} 和 S_{Cache} 的完整模型的颜色控制更加合理；对于速写风格，仅 S_{Text} 已经能够表现风格，并且内容保持良好，但是仅 S_{Cache} 的迁移结果较为杂乱。可以看出 S_{Text} 具有单独处理风格迁移的能力，而没有 S_{Text} 只有 S_{Cache} 风格迁移质量较差，说明对于常见风格，S_{Text} 为必选项。S_{Cache} 为可选部分，不能单独处理风格迁移，但对

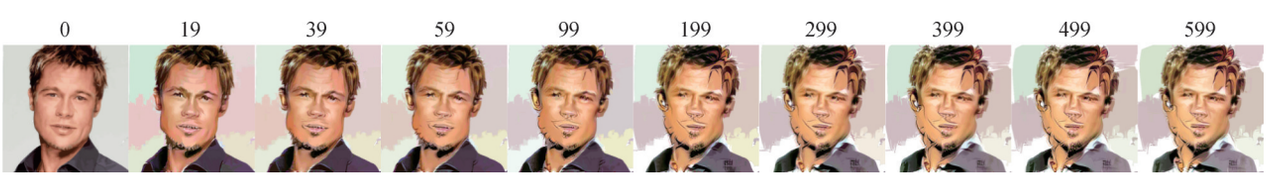

图 5　画面随迭代次数变化

Fig. 5　The variation of the screen with the number of iterations

图6　消融实验1：验证各模块的有效性

Fig. 6　Ablation experiment 1: verify the effectiveness of each module ((a) style text prompt input; (b) content vector graphic input; (c) full model; (d) without $L_{Content}$; (e) without S_{Cache}; (f) without S_{Text})

图7　消融实验2：4种风格实验验证S_{Text}和S_{Cache}的有效性

Fig. 7　Ablation experiment 2: proving the effectiveness of S_{Text} and S_{Cache} using four styles of experiments

于小众风格有较强的风格迁移强化效果。如图7速写风格所示，仅S_{Cache}生成的矢量图能够表现一定的风格，但是对内容可能有较大破坏；而对于海贼王仅S_{Text}已不足以控制风格，内容出现部分失真，此时仅有S_{Cache}而不使用S_{Text}也能表现出不错的风格迁移能力，两者结合的完整模型则大幅减少预期外的内容变化。如海贼王风格结果所示，融合两个部分的完整模型对于内容控制较好，能够削弱内容失真情况，只有S_{Cache}会使得头发变化，在仅S_{Text}的情况下，由于目前大模型仍然难以分辨人体肌肉形状，因此会出现肌肉转变为其他内容的现象，而融合S_{Cache}对这种情况有所改善。而对于水墨风格，仅S_{Cache}会使得画面内容失真，但同时使用了S_{Text}和S_{Cache}的完整模型风格效果更加明显。

3　结　论

本文提出了一种基于CLIP风格评分模型的矢量图风格迁移方法，用CLIP风格评分模型引导矢量图的风格迁移，使风格迁移更符合人的视觉感知，并用文本代替图像指定风格，更加准确和可控。本文还提出了few-shot的CLIP新风格感知，只需输入4～16幅代表新风格的图像即可实现对CLIP未见过的风格进行迁移，且无需训练。大量实验表明了本文方法的有效性，与现有的3个主流方法相比，本文方法能更好地保证风格迁移生成更加符合人类视觉的矢量图。区别于之前的方法，本文直接借助具有识别图像风格能力的CLIP模型监督矢量图形参数变化，由于CLIP识别的能力更符合人类的感觉，因此迁移的结果比之前方法更偏向全局风格，而不只是局部的纹理，实验表明了本文方法的有效性。

本文方法还存在一定的局限性，如风格迁移结果仍然有可能附带部分不属于风格的特征；与其他矢量图风格迁移方法相似，本文生成画面不够精细，放大局部会看到瑕疵。动态选择小范围的绘画区域（Hu等，2023），或许能很好地解决这一问题，这也是未来工作之一。

参考文献（References）

Efimova V, Chebykin A, Jarsky I, Prosvirnin E and Filchenkov A. 2023. Neural style transfer for vector graphics [EB/OL]. [2025-07-16]. https://arxiv.org/pdf/2303.03405.pdf

Gao P, Geng S J, Zhang R R, Ma T L, Fang R Y, Zhang Y F, et al. 2023. CLIP-Adapter: better vision-language models with feature adapters. International Journal of Computer Vision, 2024, 132(2): 581-595 [DOI: 10.1007/s11263-023-01891-x]

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 (CVPR). Las Vegas, USA: IEEE: 2414-2423 [DOI: 10.1109/CVPR.2016.265]

Hu J C, Xing X M, Zhang J and Yu Q. 2025. VectorPainter: advanced

stylized vector graphics synthesis using stroke-style priors//Proceedings of 2025 IEEE International Conference on Multimedia and Expo. Nantes, France: IEEE: 1-6 [DOI: 10.1109/ICME59968.2025.11210204]

Hu T, Yi R, Zhu H K, Liu L, Peng J L, Wang Y B, et al. 2023. Stroke-based neural painting and stylization with dynamically predicted painting region//Proceedings of the 31st ACM International Conference on Multimedia. Ottawa, Canada: ACM: 7470-7480 [DOI: 10.1145/3581783.3611766]

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 (ICCV). Venice, Italy: IEEE: 1510-1519 [DOI: 10.1109/ICCV.2017.167]

Li T M, Lukáč M, Gharbi M and Ragan-Kelley J. 2020. Differentiable vector graphics rasterization for editing and learning. ACM Transactions on Graphics, 39(6): #193 [DOI: 10.1145/3414685.3417871]

Radford A, Kim J W, Hallacy C, Ramesh A, Goh G, Agarwal S, et al. 2021. Learning transferable visual models from natural language supervision//Proceedings of the 38th International Conference on Machine Learning. [s.l.]: PMLR: 8748-8763

Rombach R, Blattmann A, Lorenz D, Esser P and Ommer B. 2022. High-resolution image synthesis with latent diffusion models//Proceedings of 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). New Orleans, USA: IEEE: 10674-10685 [DOI: 10.1109/CVPR52688.2022.01042]

Schaldenbrand P, Liu Z X and Oh J. 2022. StyleCLIPDraw: coupling content and style in text-to-drawing translation//Proceedings of the 31st International Joint Conference on Artificial Intelligence. Vienna, Austria: IJCAI Organization: 4966-4972 [DOI: 10.24963/ijcai.2022/688]

Sun M T, Dai L Q and Tang J H. 2023. Transformer-based multi-style information transfer in image processing. Journal of Image and Graphics, 28(11): 3536-3549 (孙梅婷, 代龙泉, 唐金辉. 2023. 基于Transformer方法的任意风格迁移策略. 中国图象图形学报, 28(11): 3536-3549) [DOI: 10.11834/jig.211237]

Ulyanov D, Lebedev V, Vedaldi A and Lempitsky V. 2016. Texture networks: feed-forward synthesis of textures and stylized images//Proceedings of the 33rd International Conference on Machine Learning. New York, USA: JMLR.org: 1349-1357

Zhang R, Isola P, Efros A A, Shechtman E and Wang O. 2018. The unreasonable effectiveness of deep features as a perceptual metric//Proceedings of 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Salt Lake City, USA: IEEE: 586-595 [DOI: 10.1109/CVPR.2018.00068]

Zhang Y X, Dong W M and Xu C S. 2025. Arbitrary image style transfer based on swapping attention mechanism. Journal of Image and Graphics, 30(9): 3141-3152 (张宇欣, 董未名, 徐常胜. 2025. 基于交换注意力机制的任意图像风格迁移. 中国图象图形学报, 30(9): 3141-3152) [DOI: 10.11834/jig.240652]

作者简介

郑舒洋，男，硕士研究生，主要研究方向为矢量图风格迁移。
E-mail: 1971596225@qq.com
陈佳舟，通信作者，男，副教授，主要研究方向为计算机图形学、可视分析和文化遗产数字化保护。
E-mail: cjz@zjut.edu.cn
朱欣定，男，博士研究生，主要研究方向为草图动画。
E-mail: 316454596@qq.com
李凯勇，男，教授，主要研究方向为非物质文化遗产数字化保护。E-mail: 492302219@qq.com