基于IP-Adapter的纹理合成

摘要

现有的纹理合成方法主要利用了通过某种模型从纹理图片中获取的特征，并且主要聚焦于如何生成指定大小、结构相似、没有瑕疵的纹理图片。本文主要关注如何生成外观上相似，但内部结构有所不同的纹理。简单地说，本文尝试用IP-Adapter对纹理结构进行修改，并引入kv替换技术或注意力蒸馏损失来保持纹理的外观；同时，为了更好地结合IP-Adapter带来的结构变化和原始外观，本文根据扩散模型“由粗到细”的先验，调整了外观保持技术的步数。最终，本文能够生成与参考纹理外观相似，但结构有所变化的纹理，特别是针对非稳态纹理。

关键词：纹理合成；IP-Adapter；kv替换；注意力蒸馏损失

1 引言

纹理通常定义为由某种图案重复排列、具有一定结构的图片。作为计算机图形学的一环，纹理对于CG动画、游戏和3D建模领域十分重要，能够显著提升画面的视觉表现。纹理通常由纹理艺术家借助相关工具制作而成，但这种通常效率较低，且可能会产生大小不匹配、缝隙等问题。对此的解决方案之一是进行纹理合成。与人工制作的方法相比，纹理生成具有高效、易使用、质量稳定等特点。

纹理合成方法的主要任务是，给定参考纹理，生成与之在外观上相似，且结构合理、没有明显瑕疵的指定大小纹理图片。为了实现该任务，目前主流方法通常会从参考纹理图片中提取特征，然后把这种特征作为纹理生成的依据——通常是根据该特征对纹理进行优化。由于这些方法直接使用了从参考纹理图片中提取的特征，生成结果的结构往往与参考纹理十分接近。

从开始2020年，扩散模型[1,2]逐渐成为生成模型的主流。其中DALLE2[3]模型利用了图像CLIP[4]编码，能够在没有文本提示的情况下生成与参考图像语义和外观上同时相似的图像。在那之后，其他研究者[1,2]利用CLIP图像编码对原本的扩散模型进行了微调，使得扩散模型除了文本之外，还能使用图像作为条件信息进行生成。2023年，IP-Adapter[5]被提出，该模型在不修改原始扩散模型的基础上，引入了额外的适配器，使得扩展模型能够接受图像作为生成条件，且效果相比于之前的方法有所提升。

我通过实验发现，这些利用图像CLIP编码进行生成的模型生成的纹理，在结构上与参考纹理有所区别，特别是非稳态纹理——这是目前其他纹理生成模型做不到的。但这些模型生成的纹理在外观细节上往往有所欠缺，因此，我基于IP-Adapter，尝试引入了kv替换技术或注意力蒸馏损失，还调整了外观保留技术生效的步数。最终，能够生成外观基本一致，结构有所变化的纹理图像。

[1] https://huggingface.co/lambdalabs/sd-image-variations-diffusers
[2] https://huggingface.co/stabilityai/stable-diffusion-2-1-unclip

2 相关工作

纹理合成方法可以大致分为传统方法和基于深度学习的方法。

2.1 传统纹理合成方法

传统纹理合成方法主要有三种，基于像素的方法[6,7]、基于缝补的方法[8,9]、基于优化的方法[10,11]。目前，传统方法中效果比较突出的是自调整纹理优化方法[12]和块匹配加速的方法[13,14]；前者通过合理的初始化，并引入边缘图作为指导，实现了结构连贯的纹理合成，它无法处理包含大规模但不显著特征的纹理。总的来说，由于没有参数或参数过少，传统方法生成结果的随机性往往有所欠缺，且通常只适用于结构尺度较小的纹理。

2.2 基于深度学习的纹理合成方法

第一个基于深度学习的纹理合成方法由 Gatys[15]等人提出。该方法利用 CNN 的特征提取能力，结合作者提出的 Gram Loss，不断优化生成纹理，缩小与参考纹理之间的差距，最终在外观上接近参考纹理。Gram Loss 主要计算的是特征之间 Gram 矩阵的差距，Gram 矩阵衡量了特征不同通道之间的关系，但因此也忽略了空间位置的排列。相似地， Heitz[16]等人提出了 Sliced Wasserstein Distance，用于替代 GramLoss，并从理论和实践层面证明了 SWD 的有效性与鲁棒性，实现了更好的纹理生成结果。

在那之后，一些基于GAN的方法[17,18,19,20]被相继提出。其中，NonGAN[20]在训练阶段对参考纹理进行裁剪取块，然后又从该块纹理中裁剪出宽高减半的纹理块，通过对抗学习，使得生成器在给定输入纹理的情况下，能够生成宽高扩展后的纹理。但是，该方法就像其他基于GAN的方法一样，面临着训练不稳定、边缘细节瑕疵等问题。

最近，NTSGC[21]被提出。该方法结合了传统纹理合成中的马尔科夫场[22]优化框架和现代 VGG[23]网络，实现了高质量的纹理合成，并且能够通过提出了GCD Loss，添加额外的控制条件进行纹理合成。随着扩展模型的兴起，也有研究者提出了基于扩展模型的纹理合成方案，主要的思路包括但不限于注意力中的键值、对模型进行微调等。

3 本文方法

3.1 扩散模型

扩散模型是生成模型的一种，通过对原始高斯噪声进行去噪操作来生成图片。扩散模型主要由两个过程组成：加噪过程和去噪过程。训练时，首先使用T步马尔科夫链向图像逐步添加噪声；加噪过程结束后，又逐步使用UNet预测加噪后图像中的噪声，从加噪图像中去除预测出的噪声，最终恢复一张清晰的图像。在这过程中，UNet预测出的噪声会与图像中实际添加的噪声做损失，然后进行反向传播，优化UNet网络参数。

在扩散模型中，条件控制主要通过交叉注意力引入。一般的扩散模型使用CLIP对文本进行编码，结果作为文本特征插入UNet中所有的交叉注意力中，以控制图片的生成。若用Z表示UNet中的查询特征，用f_t表示文本特征，则交叉注意力的输出为：

$$Z = Attention(Q, K, V) = Softmax(\frac{QK^T}{\sqrt{d}})V \tag{1}$$

其中，$Q = ZW_q$，$K = f_tW_k$，$V = f_tW_v$，三者分别是交叉注意力操作中的查询、键和值矩阵，W_q、W_k、W_v是可训练线性投影层的权重矩阵。

3.2 IP-Adapter

IP-Adapter方法示意图如图1所示：

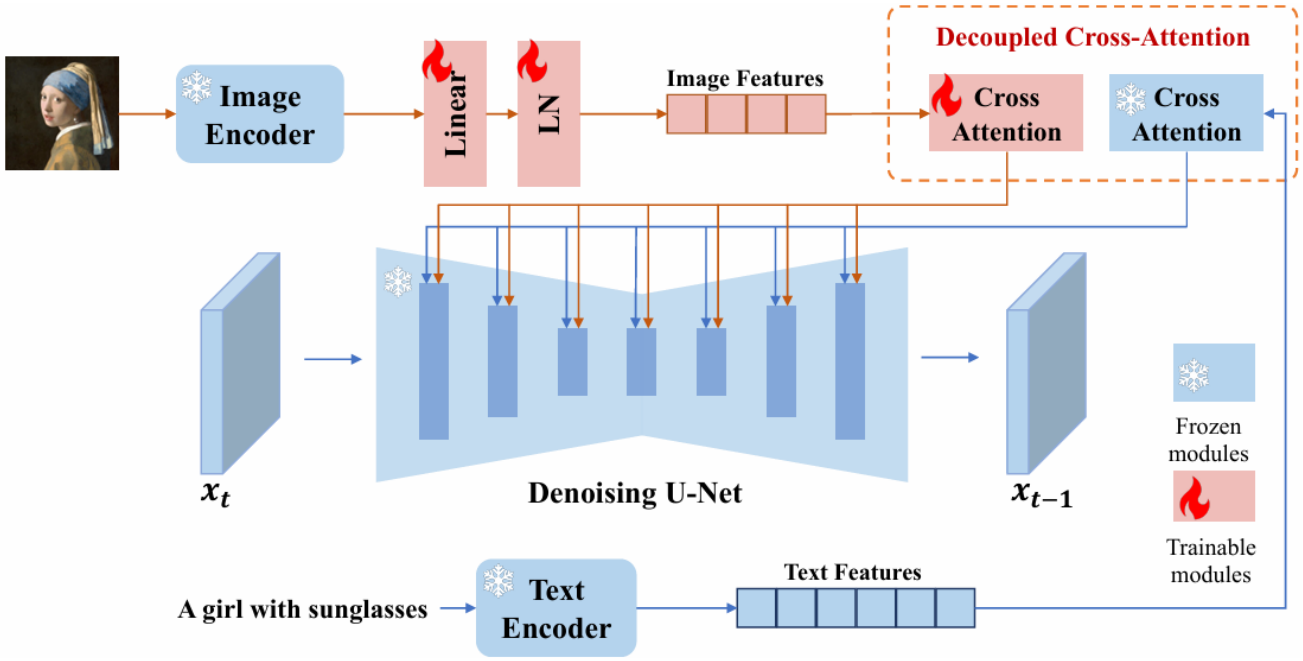

图1. IP-Adapter方法示意图

在IP-Adapter中，作者想要引入图片作为文本以外的控制条件来指导生成过程。具体来说，作者使用CLIP同时作为图像编码器和文本编码器，图像编码结果经过一个由线性层和层归一化层组成的映射网络，调整为与交叉注意力匹配的大小，与文本特征一同进入解耦交叉注意力模块。在解耦交叉注意力中，图像prompt的交叉注意力与文本prompt的交叉注意力合并，作为UNet下一环节的输入。用公式表示如下：

$$Z_{new} = Softmax(\frac{QK^T}{\sqrt{d}})V + Softmax(\frac{QK'^T}{\sqrt{d}})V' \tag{2}$$

其中，$K' = f_iW'_k$，$V' = f_iW'_v$，f_i表示图像特征。在训练时，只有映射网络和图像交叉注意力会被优化，其他模块都被冻结。

4 复现细节

4.1 与已有开源代码对比

IP-Adapter源代码和训练后的适配器都已开源，但由于直接使用IP-Adapter进行纹理生成的效果在外观上达不到预期，因此我在IP-Adapter源代码的基础上，分别尝试引入两种不

同的外观保留技术，以及调整了插入外观保留技术的时间步。

4.2 实验环境搭建

本文实验环境参考IP-Adapter原文，显卡为单张的Nvidia RTA 6000 Ada。

4.3 使用说明

本文代码包括了IP-Adapter源码、我个人复现的同组工作ADLoss代码、IP-Adapter结合kv替换的代码、IP-Adapter结合ADLoss的代码，每个代码包含在一个独立的jupyter笔记本内，可直接运行。此外，运行所需要的参考纹理保存在“纹理”文件夹内，结果保存在“outputs”文件夹内，IP-Adapter的适配器参数保存在“models”文件夹内。

4.4 创新点

本文首先尝试在IP-Adapter的基础上加入kv替换技术，以达到保持生成纹理外观的目的。Kv替换技术指的是，在去噪时，同时对用反演技术得到的参考图像噪声和生成用的随机高斯噪声去噪，并把参考图像去噪产生的kv替换生成图像去噪产生的kv。这种基于kv的方法在各种需要保持外观的研究中都有用到，本文参考MasaCtrl[24]，只在UNet的最后两个上采样块进行kv替换。同时，本文还根据“扩散模型去噪是一个由粗到细的过程，可分为三阶段，分别决定了生成图像的布局、形状和外观细节”的先验，只在去噪过程的约最后三分之一部分加入kv替换技术，在实际生成时，该参数可能需要调整。修改后的IP-Adapter结合kv替换技术的框架如图所示：

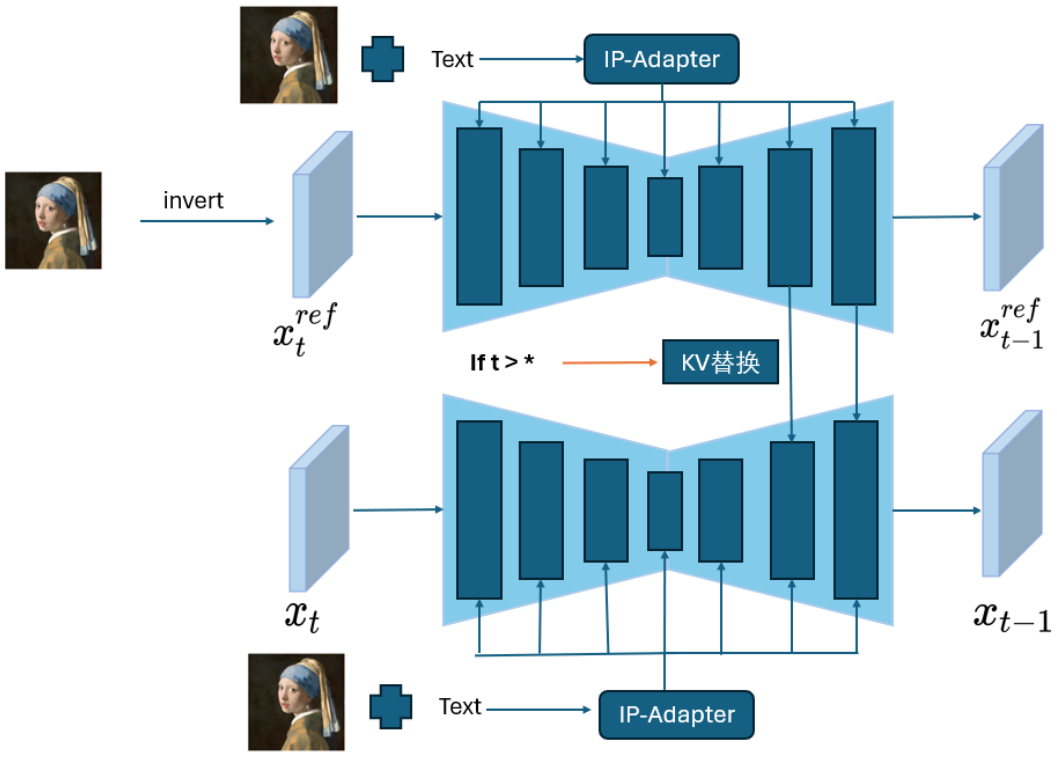

图2. IP-Adapter结合kv替换示意图

除了kv替换，本文还尝试把组内的最新工作注意力蒸馏损失,简称AD Loss，加入到IP-Adapter中。AD Loss的方法如图xx所示，该方法在把参考图像在去噪时产生的kv与噪声图像去噪时产生的q结合，计算出注意力A，然后把注意力A与去噪图像的注意力做L1损失，用于优

化扩展模型去噪后的噪声。该方法能够有效地把参考图像外观迁移到生成图像中，除了纹理合成，还能完成风格迁移、指定风格下的文生图、外观迁移等任务。

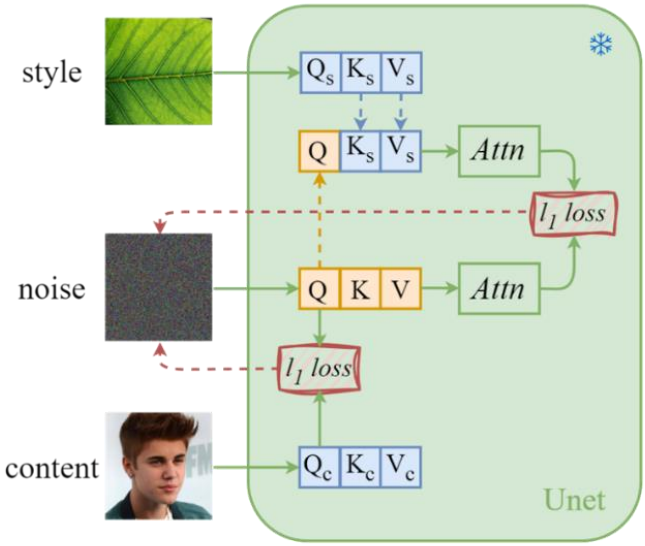

图3. AD Loss示意图

把IP-Adapter和AD Loss结合后的框架图如图4所示

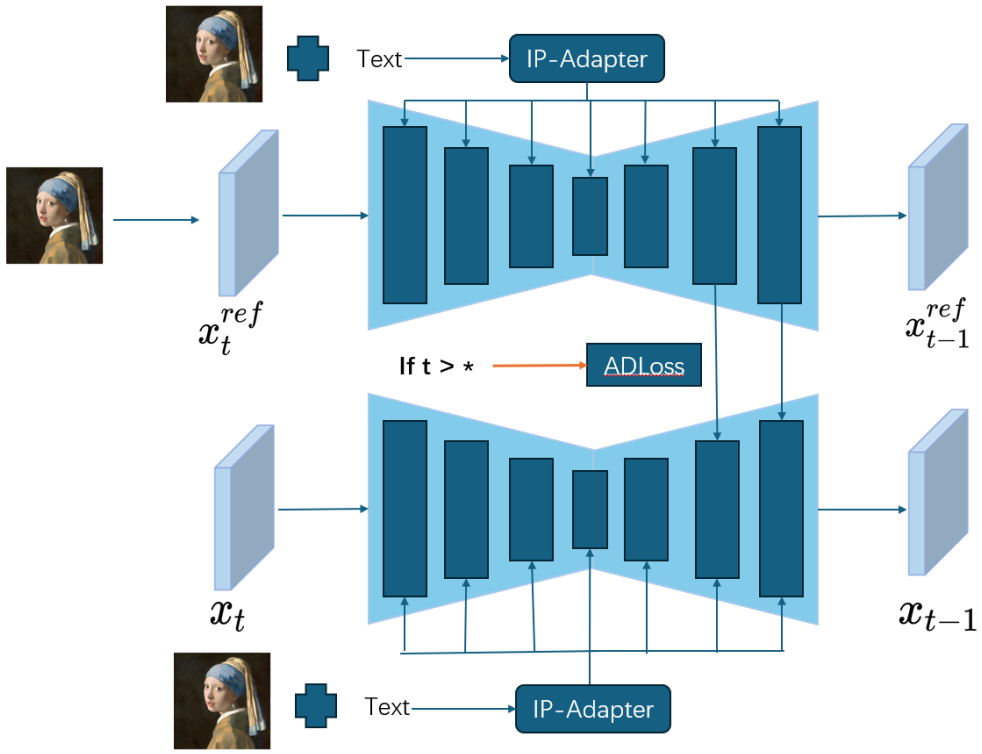

图4. IP-Adapter结合AD Loss替换示意图

5 实验结果分析

图5第一行展示了参考纹理，剩下三行分别展示了IP-Adapter、IP-Adapter结合kv替换、IP-Adapter结合AD Loss的生成结果，实验全程使用相同的随机种子噪声，步数控制在50步，统一在去噪的最后三分之一部分插入外观保留技术。可以看到原始IP-Adapter生成结果在结构上与参考

纹理有所区别，但外观上的差别较大；插入外观保留技术后，生成的结果较好的结合了IP-Adapter带来的结构变化和参考纹理的外观。

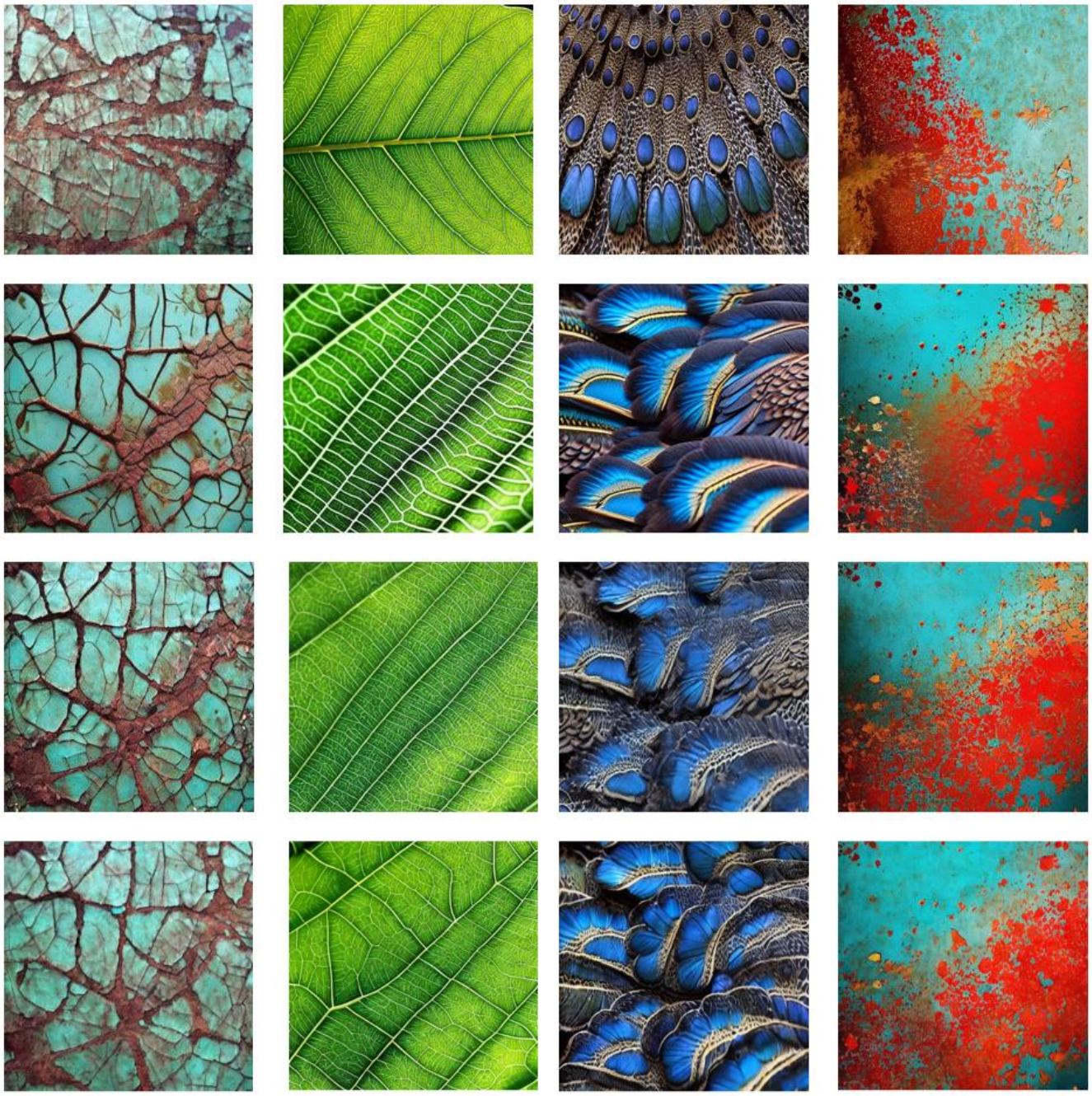

图5.生成结果

6 总结与展望

本文通过实验发现，IP-Adapter能够生成相比于参考纹理结构有所变化的结果，但外观保留上效果不达预期。因此，本文引入了kv替换技术或注意力蒸馏损失来使，同时还调整了两个外观保留技术插入的时机，得生成结果能够在符合IP-Adapter带来的结构变化的同时保持参考纹理的外观。

参考文献

[1] Ho J, Jain A, Abbeel P. Denoising diffusion probabilistic models[J]. Advances in neural information processing systems, 2020, 33: 6840-6851.

[2] Rombach R, Blattmann A, Lorenz D, et al. High-resolution image synthesis with latent diffusion models[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2022: 10684-10695.

[3] Ramesh A, Dhariwal P, Nichol A, et al. Hierarchical text-conditional image generation with clip latents[J]. arXiv preprint arXiv:2204.06125, 2022, 1(2): 3.

[4] Radford A, Kim J W, Hallacy C, et al. Learning transferable visual models from natural language supervision[C]//International conference on machine learning. PMLR, 2021: 8748-8763.

[5] Ye H, Zhang J, Liu S, et al. Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models[J]. arXiv preprint arXiv:2308.06721, 2023.

[6] Efros A A, Leung T K. Texture synthesis by non-parametric sampling[C]//Proceedings of the seventh IEEE international conference on computer vision. IEEE, 1999, 2: 1033-1038.

[7] Wei L Y, Levoy M. Fast texture synthesis using tree-structured vector quantization[C]//Proceedings of the 27th annual conference on Computer graphics and interactive techniques. 2000: 479-488.

[8] Efros A A, Freeman W T. Image quilting for texture synthesis and transfer[M]//Seminal Graphics Papers: Pushing the Boundaries, Volume 2. 2023: 571-576.

[9] Kwatra V, Schödl A, Essa I, et al. Graphcut textures: Image and video synthesis using graph cuts[J]. Acm transactions on graphics (tog), 2003, 22(3): 277-286.

[10] Kwatra V, Essa I, Bobick A, et al. Texture optimization for example-based synthesis[M]//ACM Siggraph 2005 Papers. 2005: 795-802.

[11] Wexler Y, Shechtman E, Irani M. Space-time completion of video[J]. IEEE Transactions on pattern analysis and machine intelligence, 2007, 29(3): 463-476.

[12] Kaspar A, Neubert B, Lischinski D, et al. Self tuning texture optimization[C]//Computer Graphics Forum. 2015, 34(2): 349-359.

[13] Barnes C, Shechtman E, Finkelstein A, et al. PatchMatch: A randomized correspondence algorithm for structural image editing[J]. ACM Trans. Graph., 2009, 28(3): 24.

[14] Darabi S, Shechtman E, Barnes C, et al. Image melding: Combining inconsistent images using patch-based synthesis[J]. ACM Transactions on graphics (TOG), 2012, 31(4): 1-10.

[15] Gatys L, Ecker A S, Bethge M. Texture synthesis using convolutional neural networks[J]. Advances in neural information processing systems, 2015, 28.

[16] Heitz E, Vanhoey K, Chambon T, et al. A sliced wasserstein loss for neural texture synthesis[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021: 9412-9420.

[17] Jetchev N, Bergmann U, Vollgraf R. Texture synthesis with spatial generative adversarial networks[J]. arXiv preprint arXiv:1611.08207, 2016.

[18] Bergmann U, Jetchev N, Vollgraf R. Learning texture manifolds with the periodic spatial GAN[J]. arXiv preprint arXiv:1705.06566, 2017.

[19] Shaham T R, Dekel T, Michaeli T. Singan: Learning a generative model from a single natural image[C]//Proceedings of the IEEE/CVF international conference on computer vision. 2019: 4570-4580.

[20] Zhou Y, Zhu Z, Bai X, et al. Non-stationary texture synthesis by adversarial expansion[J]. arXiv preprint arXiv:1805.04487, 2018.

[21] Zhou Y, Chen K, Xiao R, et al. Neural texture synthesis with guided correspondence[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023: 18095-18104.

[22] Kwatra V, Essa I, Bobick A, et al. Texture optimization for example-based synthesis[M]//ACM Siggraph 2005 Papers. 2005: 795-802.

[23] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition[J]. arXiv preprint arXiv:1409.1556, 2014.

[24] Cao M, Wang X, Qi Z, et al. Masactrl: Tuning-free mutual self-attention control for consistent image synthesis and editing[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2023: 22560-22570.