Text2Mesh: Text-Driven Neural Stylization for Mesh

摘要

在这项工作中，作者开发了用于编辑三维对象样式的直观控件。Text2Mesh 通过预测符合目标文本提示的颜色和局部地理测量细节来对 3D 网格进行风格化。使用固定网格输入（内容）与人工神经网络耦合的 3D 对象的解纠缠表示，我们称之为神经样式场网络(NSF)。为了修改样式，通过利用 CLIP 的代表性来获得文本提示（描述样式）和样式化网格之间的相似性分数。Text2Mesh 既不需要预先训练的生成模型，也不需要专门的三维网格数据集。它可以处理具有任意亏格的低质量网格（非流形、边界等），并且不需要 UV 参数化。

关键词：CLIP；NSF

1 引言

该文章提出了一个文本驱动生成风格化 3D 网格的框架，与 text-to-image 都是文本驱动生成，并且文章详细的介绍了工作流程和创新技术。使用神经网络模型基于 MLP 模型，通过 MLP 编码后，分别用于生成颜色和法线的输出。并且文章提出不需要进行预训练模型和专门的数据，对硬件要求较低，单 CPU 也能够训练，短时间之内就能生成符合语义的结果。本文实验部分采用的对比是根据在不同情况下的 CLIP 的相似度得分进行比较和分析，就所有优化和增强措施进行分析，最后得到一个最优的模型。同时也解释了某些情况得分低的原因，以及某些不符合文本提示但是得分高的现象。

2 相关工作

与本文相关的其他工作包括基于 CLIP 的文本驱动和三维模型的样式转换。对于三维网格样式的各个方面可以通过网格参数化对表面纹理进行修改控制，并且避免的对输入网格的参数化，使用神经场来修改样式，即用 RGB 颜色和位移，同时还考虑了颜色和局部地理的位置。作者的工作包括利用神经网络的归纳偏差完成图像去噪、曲面重建、图像合成和编辑等，利用神经网络的诱导偏差为先验，引导 text2mesh 远离 CLIP 嵌入空间中的退化方案，也就是利用位置编码来合成细粒度的纹理细节。

2.1 文本驱动

本文的工作通过 CLIP 嵌入文本描述，对图像进行处理。CLIP 用于引导图像的生成，并生成风格化的图像。同时进行的工作包括使用 CLIP 微调预训练的 StyleGAN，利用 ShapeNet 数据集和 CLIP 进行无条件三维图像生成。第一个利用 CLIP 进行合成而不需要预训练的网络或者数据集是 CLIPDraw，用于文本引导生成二维矢量图形。本文通过对风格化的 Mesh 进行分解成多个视角，对每个视角进行多次 2D 增强渲染，并基于 CLIP 的语义损失对目标进行计算相似度得分。

2.2 三维中的几何样式转换

有些方法对三维的几何物体进行分析，找出相似的集合元素和风格不同的部分；其他方法根据内容不同来区分和传递几何风格。mesh-renderer 在目标图像的驱动下改变源网格的颜色和几何形状。有的方法通过添加地理度量细节改变三维形状。本文提出的方法在直观简洁的文本规范指导下进行，并且考虑了多种样式。三维网格样式的各个方面可以通过网格参数化对表面纹理进行控制，而本论文完全避免了参数化，采用神经场修改样式，即通过 RGB 值和偏移进行操作，这样能够避免参数化。

一些将几何图形和外观表现纠缠在一起的神经场，限制了对内容和风格的可分离控制，通常难以描述尖锐的特征细节，渲染速度慢，编辑操作困难。本文完全相反，使用明确的网格形状控制法和控制外观的神经样式场，对三维物体进行分解表示，这样既能避免参数化，又能处理外观和生成高分辨率输出。

3 本文方法

3.1 本文方法概述

Text2Mesh 通过预测颜色和几何细节来修改输入网格以符合目标文本。通过绘制多个 2D 图像和应用 2D 增强来优化神经风格网络的权重，这些增强从基于 CLIP 的语义模型中获得与目标的相似性得分。将 Mesh 上的每个点 P 进行标准化的编码，在经过神经网络渲染的时候生成两条通道，RGB 颜色和位移，用于预测图像生成。

工作流程如 1所示：

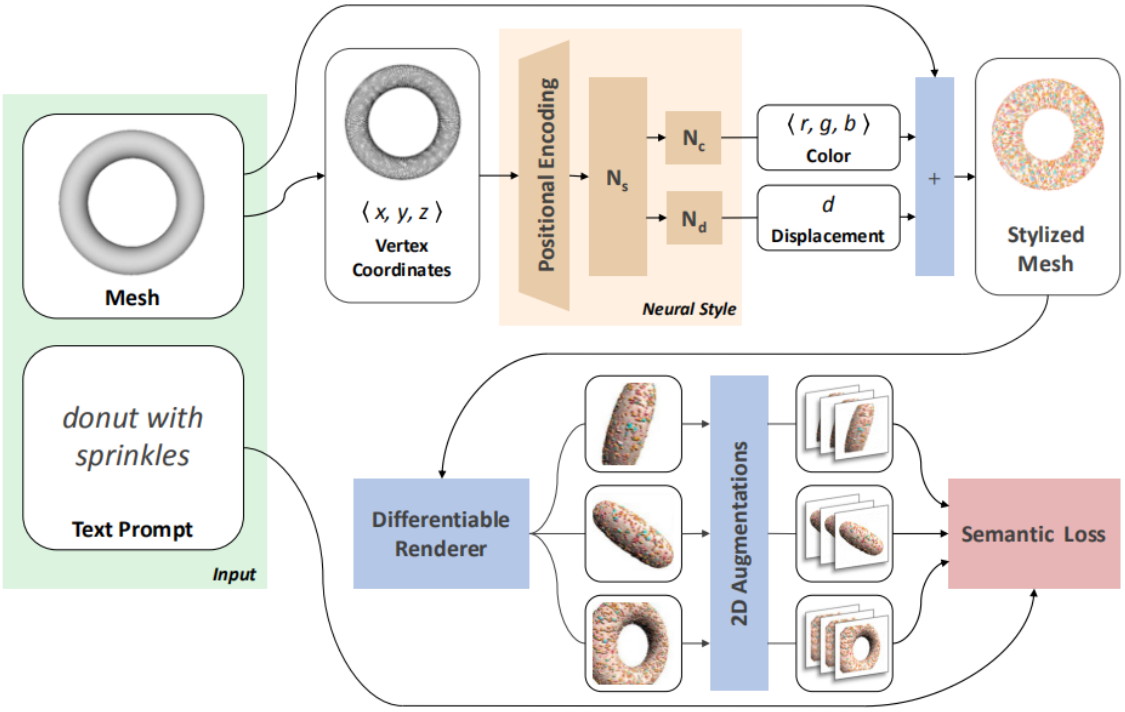

图 1. 方法示意图

3.2 神经样式场网络

本文的神经网络是 NSF(Network Style Field), 通过为每个顶点预测一个样式属性，从而在整个形状表面定义样式场。NSF 由 MLP 编码构成，它将网格上的点 P 映射为沿着指定的表面法线的颜色和位移。通过预测颜色和位移这种方式让样式场和原网格紧密耦合，从而能够实现局部的几何修改和细粒度修改。如果只预测颜色的话，会导致 NSF 用阴影对局部几何细节进行幻化，生成的风格化网格质量就很一般，所以还需要加入沿着表面法线的位移。同时这种方式还能兼顾局部几何细节和全局感知，不会导致局部细节与文本提示或者全局感知脱离。用 NSF 对三维物体进行分解表示，通过多个二维视图渲染三维风格化网格，引导 NSF 网络。使用明确的网格形状表示法和控制外观的 NSF，避免参数化。

NSF 使用低纬坐标作为 MLP 的输入，这样会表现出频谱偏差。为了合成高频的细节，本文在位置编码的时候使用了傅里叶特征映射，使 MLP 能够消除频谱偏差并学习插值高频函数。

3.3 CLIP 得分分析

去除样式场网络（−net），而直接优化顶点颜色和位移，导致表面上的噪声和任意位移，随机二维增强是生成有意义的剪辑引导绘图的必要条件。即删除二维增强会导致与目标文本提示符完全无关的程式化; 如果没有傅里叶特征编码（−FFN），生成的样式将失去所有细粒度的细节; 去掉 crop 增强后，输出同样无法合成定义目标的细粒度样式细节; 去除损失函数的几何组件（−disl）会阻碍几何细化，网络通过阴影模拟几何进行补偿，而网络则会通过阴影模拟几何来进行补偿。如果没有几何先验（−3D），就没有源 mesh 来给定全局结构，因此，在三维空间中的二维平面被视为一个图像画布。

可以看出当没有进行 2D 增强的时候得分最低，因为相似性得分是从多次 2D 增强的基于 CLIP 的语义损失获得，而不采用傅里叶编码就无法插入高频细节，导致无法合成细节。虽然没有几何先验，但是该测试在相似性得分上仍然能取得较高分，虽然完全看不出原来的形状，但是细节方面能够合成。最终取得不同消融下的最高得分。

CLIP 在不同情况下的得分如 2所示：

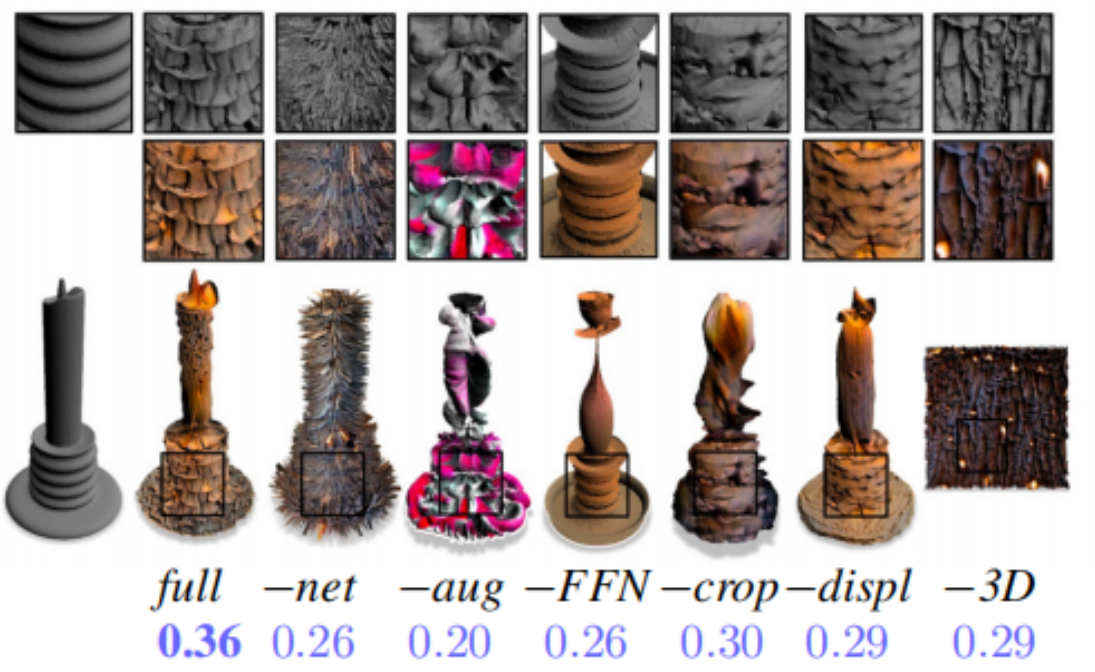

图 2. CLIP 得分

4 复现细节

4.1 与已有开源代码对比

本文在 github 开源代码，搜索 text2mesh。本篇论文没有提出很复杂的神经网络，主要是提出了一种新型的框架。本框架基于 MLP 编码，利用一些更新 Mesh 和计算损失的函数进行更新网格，通过计算近 100 次的损失来渲染，在神经风格场内用基于法线的位移对图像进行细节生成。本次复现与开源代码相比没有较大改动，在性能方面也没有明显的改善。

4.2 实验环境搭建

本文的实验环境在 Ubuntu18.04，Linux 内核版本 5.04，CUDA 版本为 11.4，GPU 为 Tesla-V100，pytorch 版本为 1.12.1。其余环境配置如下：python=3.9，torchvision=0.13.1，torchaudio=0.12.1，cudatoolkit=11.3，matplotlib=3.5.2，jupyter=1.0.0，pip=21.1.2，以及 CLIP 和 Kaolin。

4.3 使用说明

通过 git clone 或者直接下载源码，在代码目录下的/demo 里有多个脚本文件，先通过 chmod +x xxxx.sh 赋予 shell 文件权限，在文件内修改参数，包括原目录、目标目录、原网格

和文本描述等。修改 prompt 和 objpath 用于生成不同的风格化网格。当所以前置条件都准备好之后，运行 sh 文件，能够看到当前渲染的进度条，以及当前渲染的语义损失生成。运行过程大约 30 分钟生成一个完成的目标风格化网格，在过程中会逐渐生成中间图片和中间模型，以及当前进行预测和渲染得到的语义损失。可以用在线查看工具打开 obj 文件查看 3D 图像。

图 3 是其中一个运行脚本的部分说明，运行时修改原始目录和目标生成目录，以及文本提示词即可。

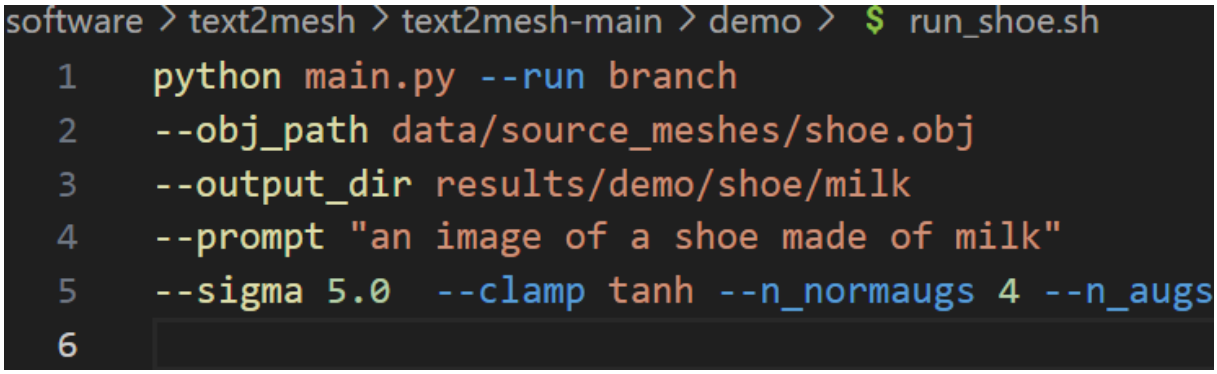

图 3. 操作界面示意

5 实验结果分析

运行过程大约 30 分钟，每隔几分钟就会生成一组当前渲染的图像，基本上十分钟就可以获得与文本描述相符的图像。通过不断用 2D 图像进行渲染，最后生成 3D 风格化网格图像。在生成高质量图像后，后面的每组图像都显示出了不同的细节，最后由这些细粒度的调整组成目标图像。以下是展示其中一个复现结果，输入的文本与源网格相关，可以得到一个合理的结果。但是如果输入与源网格无关的提示之后，也许会导致嵌入文本的时候完全抹去网格的原形状，例如：像仙人掌的仙人掌和像鸭子的猫等等提示词，这虽然说明本模型具有很大的自由性，但同时也会带来一些不可解释的结果。或者当输入 “a image of vase made of cows”，会生成一个外观上全是凸起的牛头的花瓶，这虽然在一定程度上也符合语义，但是对自然语言处理还是存在先天缺陷，作者采取的优化措施在匹配文本发生错误的时候，只需要在文本提示中加入对象类别，就可以在目标中加以保留原网格的一些描述。并且本模型无法生成一些逼真的质感，只能在大体方向上向文本进行靠近，生成较符合文本提示的风格化目标。

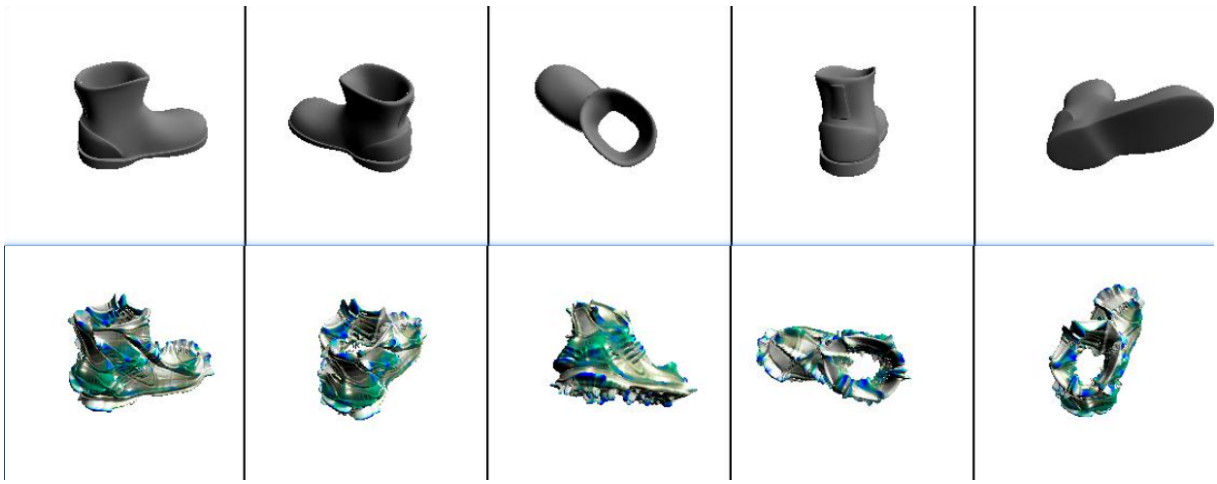

图 4. 实验结果示意

6 总结与展望

本篇论文实现了文本驱动转换成风格化的网格图像，通过 CLIP 的文本嵌入能力完成，实现的功能也很有创意，同时又没有引入很复杂的神经网络。本框架使用预先训练好的 CLIP 模型，该模型已经被证明包含偏差 [1]。在生成目标 3D 图像时，仍然存在一些缺陷，例如无法生成逼真的金属色泽、质感；细节塑造的太多，物体表面经常是凹凸不平的，对于一些表面光滑的物体生成效果可能欠佳；以及输入与原网格无关的文本提示时，可能会导致原网格的特征被抹去，生成无关的图像。在未来的研究中，可以继续研究使生成更多的风格，能够调整的参数更多样化和个性化。

参考文献

[1] Jack Clark Alec Radford Jong Wook Kim Sandhini Agarwal, Gretchen Krueger and Miles Brundage. Evaluating clip: towards characterization of broader capabilities and downstream implications. In *Proc. IEEE/CVF Conf. on Computer Vision & Pattern Recognition*, 2021.