

为纹理迁移和生成任务学习语义对齐的 UV 映射

摘要

现有的三维模型 UV 映射算法不会考虑模型之间的语义一致性，往往会将不同模型的同一语义部件映射到 UV 空间中的不同区域，使得后续的纹理迁移和生成非常困难。本文作者以特征脸技术为启发，通过简单的点云重建任务，为同一个类别中的不同三维模型重新学得了语义一致的 UV 映射，大大降低了纹理迁移和生成的难度。由于方法一直未开源，故本项目从头复现了文章中提出的算法，包括数据处理、网络实现、损失计算等，复现完成后，在纹理迁移任务上进行了验证，结果基本达到原文水平，并进一步对复现过程中发现的问题提出了新的解决方案。

关键词：UV 映射；纹理迁移；纹理生成；几何处理

1 引言

在计算机图形学领域，三维模型的纹理生成任务一直是一个很难解决的问题，其主要原因在于三维模型 UV 展开的无序性。在同一个类别中，不同三维模型语义相同的部件会被现有算法展开到 UV 空间中的不同区域，这种不一致性导致现有的图像处理和生成算法很难直接适用。为了解决这一问题，英伟达联合西蒙莎菲大学、多伦多大学、向量学院提出了 *AUV-Net: Learning Aligned UV Maps for Texture Transfer and Synthesis*，该项工作发表在计算机视觉和模式识别顶级大会 CVPR 2022 上。该工作提出了一种自动为三维模型学习语义对齐 UV 映射的方法，作者以特征脸技术为启发，通过简单的点云重建任务，学习了语义一致的三维映射，这种映射方法会将不同三维模型中语义相同的部件映射到 UV 空间中的相同位置处。

通过重新学习的三维映射，可以获得三维模型对应的对齐纹理贴图，后续进行三维模型纹理迁移的时候，只需要直接进行纹理贴图的替换即可，此外，这种语义一致的 UV 映射可以极大的降低后续进行纹理生成的难度，在作者在文章中提出，只需要简单的自编码器结构就可以实现较高质量的纹理生成。

2 相关工作

2.1 基于模板网格的纹理迁移

基于模板的纹理迁移方法假设所有的目标形状都可以通过一个可变形的模板网格表示，这种模板网格通常是一个球体 [1, 4, 6] 或者平面 [9, 10]。模板网格的 UV 需要提前计算并和模板网格绑定，在模板网格进行变形后，UV 映射会随之进行对应变形。然而，这种方法通过可

变形的网格建立源三维模型和目标三维模型之间的对应关系，由于网格的表达力受限，往往无法捕捉复杂拓扑结构的细节。例如，将一个球体变形成人体时，很难准确地重构手指，即使网格变形成功，纹理也通常无法准确地跟随变形，在目标形状上产生良好的变形效果。此外，由于模板形状先验的限制，此类方法通常不能适应不同形状的纹理和生成需求。

2.2 基于给定 UV 映射的纹理迁移

基于给定 UV 映射的纹理迁移方法假设 UV 映射和三维模型一一对应并一起提供 [3, 11]。UV 映射和纹理通常由艺术家创建，并且可以创建的时候设计任意布局。为了在这样的设定下进行纹理迁移，需要首先对三维模型和纹理贴图进行准确的语义分割，然后通过语义感知的方法进行对应的纹理生成。然而，同时对模型和贴图进行分割需要一种比较鲁棒的语义分割方法，如果分割不够准确，也会在很大程度上影响纹理迁移和生成的质量。

2.3 基于三维对应的纹理迁移

在三维模型之间进行纹理迁移的另外一个思路是先在三维模型之间建立稠密的对应关系，并在对应关系的基础上，将源三维模型中表面点的颜色直接迁移到目标三维模型对应点上。目前已经有一系列的工作在三维模型之间建立稠密对应关系 [5, 8]，不过他们在建立对应关系的时候往往只考虑几何信息，并没有把点的颜色信息考虑在内，导致直接使用对应关系进行纹理迁移出现不真实的扭曲等情况。

3 本文方法

3.1 方法概述

为三维模型学习语义对齐的 UV 映射是一个比较困难的任务，本文以特征脸技术为启发，将三维模型表示为一组对齐基的叠加，通过点云重建任务，为三维模型学习到了语义对齐的 UV 映射，该方法的总体流如图 1 所示。在进行点云重建时，考虑到三维模型表面的颜色分布通常具有区域性，即很大一块经常是同一个纯色，因此该方法同时使用三维模型空间和法向信息作为额外的约束，提升网络的学习能力。

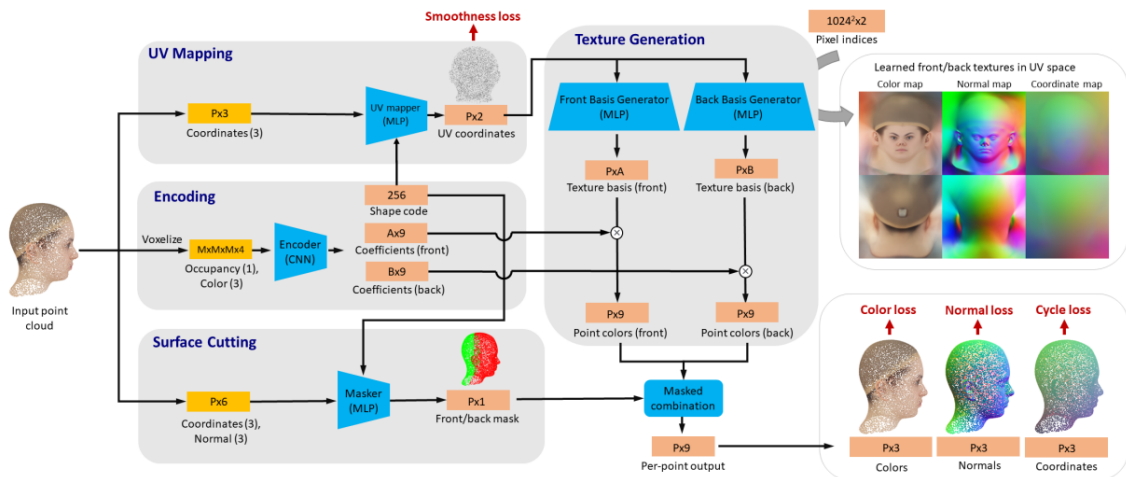


图 1. 方法示意图

3.2 网络细节

该方法包含四个不同的网络部件，分别形状编码器、形状分割器、UV 映射器以及基底生成器。输入数据会首先经过形状编码器得到形状编码和基底系数，随后，将形状编码和点云的坐标、法线输入形状分割器获得形状分割；将形状编码和坐标输入 UV 映射器获得 UV 坐标。之后，将 UV 坐标输入基底生成器生成两组基。最后，将基底系数和基进行加权累加以及掩码融合，得到最终重建的结果。

其中，形状编码器使用体素化层将输入点云体素化，之后使用三维卷积网络进行特征提取。形状分割器、UV 映射器、基底生成器均使用多层感知机结构，网络结构细节如图 2 所示。

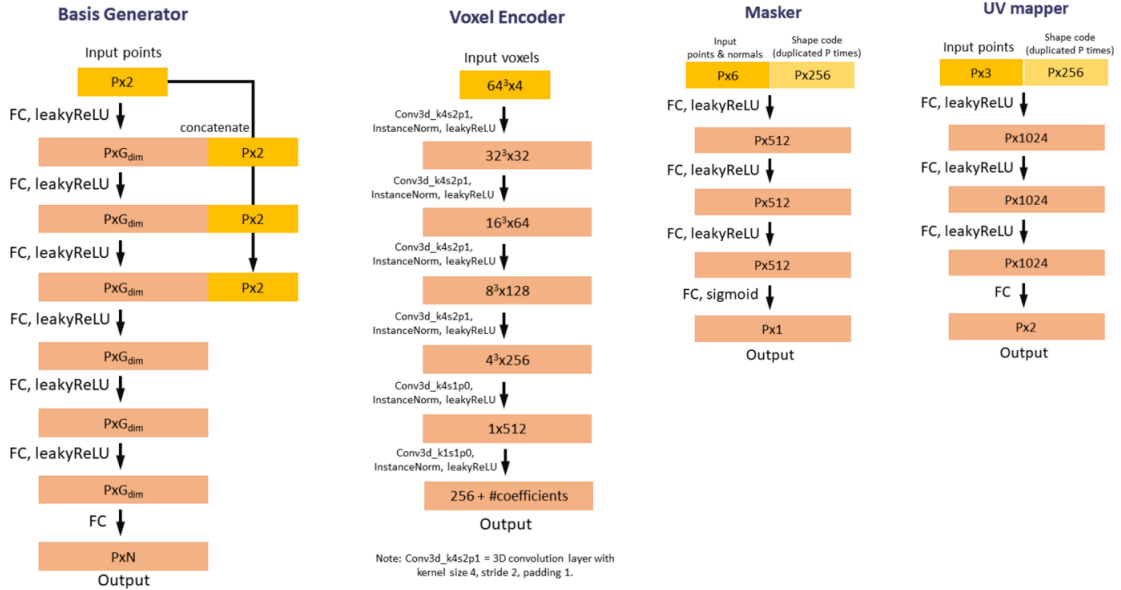


图 2. 网络细节图

3.3 损失函数

本文方法使用的损失函数主要包含两类，其中一类用来约束点云重建结果，另一类用来保证 UV 映射的合理性，具体的损失函数定义为：

$$\mathcal{L} = w_c \mathcal{L}_c + w_n \mathcal{L}_n + w_x \mathcal{L}_x + w_s \mathcal{L}_s + w_p \mathcal{L}_p \quad (1)$$

其中， \mathcal{L}_c ， \mathcal{L}_n 和 \mathcal{L}_x 为第一类损失，用来保证重建结果的正确性，直接通过计算重建结果和真实值的差距获得； \mathcal{L}_s 和 \mathcal{L}_p 为第二类损失，用来保证 UV 映射的合理性， \mathcal{L}_s 用来保证学习到的 UV 映射添加到三维模型表面时具有较小的扭曲， \mathcal{L}_p 用来为初始的 UV 映射和 UV 分割提供先验引导，其计算方法分别为：

$$\mathcal{L}_s = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N |D(p_i, p_j) - D(q_i, q_j)| \cdot T(p_i, p_j) \quad (2)$$

$$\mathcal{L}_p = \frac{1}{N} \sum_{i=1}^N (p_i^x - q_i^x)^2 + (p_i^y - q_i^y)^2 + (m_i - n_i)^2 \quad (3)$$

4 复现细节

4.1 与已有开源代码对比

截止到该项目完成时，本文工作仍未开放源码，故本次复现工作未参考任何相关源码，该项目的目标也是复现现有的方法，使用相同的数据集进行测试评估复现效果，发现该方法存在的问题并提出可能的解决方案。

4.2 网络实现

该项目使用深度学习技术，相关神经网络的搭建均基于 pytorch 和 pytorch-3D 深度学习库，各个网络模块的架构参考了作者在文章中的描述，在复现的过程中，我发现如果完全按照作者在文章中使用的网络参数实现网络，偶尔会出现特征维度不统一，导致前向传播中断的情况。因此，我重新调整了相应网络的参数设置，成功复现了原文中使用的网络架构和损失函数，并对每个网络模块以及损失函数进行了全面的测试和验证，确保其在各种输入情况下的稳定性和准确性。

4.3 数据准备

本次实验使用的数据集和作者在原文中使用的一致，均为 ShapeNet 数据集 [2]，为了由易到难的进行实验复现结果，从 ShapeNet 数据集中选了两个类别进行训练，这两个类别分别为柜子和椅子，其中柜子类内的形状差异性较小，可以简单地看作不同尺寸的立方体，网络的学习难度较小；而椅子类内的形状差异较大，网络的学习难度较大，使用的柜子和椅子的具体数目分别为 1121 个和 4564 个。

确定数据集之后，需要对数据集进行处理获得网络的训练数据。该网络需要的训练数据为带有颜色和法向信息的表面点云，为了避免采样到物体内部表面上的点，我使用了虚拟扫描方法。具体的方法为：首先在物体周围的单位球面上均匀采样 100 个点放置相机，之后在对应的位置让相机对准三维模型进行渲染，同时记录渲染出来的物体表面颜色信息、法向信息、深度信息和相机参数，接着通过空间解算得到每一个相机位置对应的局部点云，最后将局部点云进行拼接，得到完整的最终点云。

4.4 网络训练

准备好训练数据后，可以开始对网络进行训练，由于网络训练过程中并未显式的添加对应关系学习的约束，所以在网络训练的过程中，需要分为多个阶段，通过调整不同的权重来控制网络训练的目标，这里我使用了三个阶段对网络进行训练，三个阶段训练的目标分别为：减少映射形变，学习对应关系以及减少对应形变。在训练过程中我还发现，空间重建损失 \mathcal{L}_x 和平滑损失 \mathcal{L}_s 对减少映射形变影响较大，而颜色重建损失 \mathcal{L}_c 和法向重建损失 \mathcal{L}_n 对学习对应关系影响较大，网络在两个类别柜子/椅子上训练使用的超参数如表 1 所示。

表 1. 网络训练超参数表

训练阶段	训练目标	Epoch	优化器	学习率	损失权重				
					先验	平滑	空间	颜色	法向
1	减小映射形变	200/100	Adam	2e-4	10/20	10	100/10	10	10
2	学习对应关系	1000	Adam	1e-4	0	10	10/1	1	1
3	减小映射形变	800/900	Adam	1e-5	0	10	100/10	10	10

5 实验结果分析

本部分对实验所得结果进行分析，将首先介绍实验环境，之后将详细介绍语义对齐纹理贴图的获取和结果，以及使用对齐的纹理贴图在三维模型之间进行纹理迁移的结果。

5.1 实验环境

实验进行时采用的环境配置如表 2 所示，在配置运行环境时，需要先服务器上安装 Docker（可选，Docker 更加方便环境的迁移），之后，在系统中安装 Anaconda 进行 Python 运行环境的配置即可。

表 2. 环境配置表

环境配置	
操作系统	Ubuntu 20.04
开发平台	Docker
CPU	12th Gen Inter(R) Core(TM) i9-12900K 用于生成训练数据
GPU	Nvidia GeForce RTX 4090 用于模型的训练
编程语言	Python
深度学习框架	Pytorch / Pytorch-3D

5.2 语义对齐纹理贴图

基于网络重新学得的语义对齐的 UV 映射，可以为每一个三维模型重新获得语义对齐的纹理贴图，纹理贴图的获得方法是：首先将预先采样的点云输入训练好的网络中，获得点云中每一个点的分割和 UV 坐标；之后，按照预测的 UV 坐标，将点云上的颜色映射到 UV 空间，获得初始纹理贴图；最后，对初始纹理贴图进行补全，得到完整的纹理贴图。

图 3 和图 4 展示了复现代码得到的语义对齐的纹理贴图。在图 3 中，每一栏最左边展示的是初始模型的渲染图，中间展示的是补全之前的对齐纹理贴图，最右边展示的是补全之后的纹理贴图。通过观察可以发现，复现算法可以准确找到不同柜子之间的语义对应关系，得到比较合理的对齐结果。因为柜子的结构较为简单，所以我把柜子简单分成了前后两块，对于结构差异比较大的三维模型，复现方法仍然可以获得正确的对齐结果，如图 3 中第一栏第三行较高的柜子和第二栏第一行比较扁平的柜子，它们相同语义的部件（如柜门、柜顶等）都被映射到 UV 空间中的相同位置处。相应地，在图 4 中，复现方法在椅子类别上获得的纹理贴图也隐含了很强的语义一致性。图 4 中展示了三栏结果，每一栏左边展示的是初始模型的渲染图，右边展示的是补全之前的纹理贴图。由于椅子的结构相对来说更加复杂，所以我把它按照上下前后进行划分，得到了包含四个部分的纹理贴图。通过观察结果可以发现，即使是对于结构差异比较大的三维模型（如第二栏第二行和第三栏第一行），它们相同的语义部件（如椅背、椅坐、椅腿等）被较好对齐到了 UV 空间中。

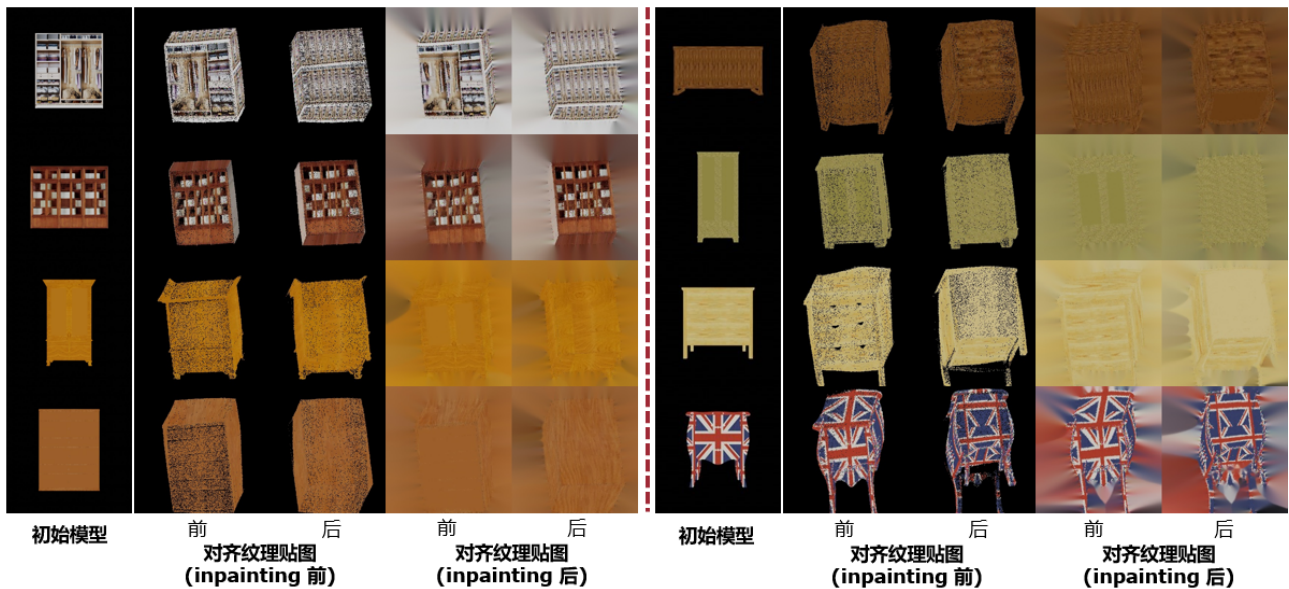


图 3. 语义对齐纹理贴图-柜子

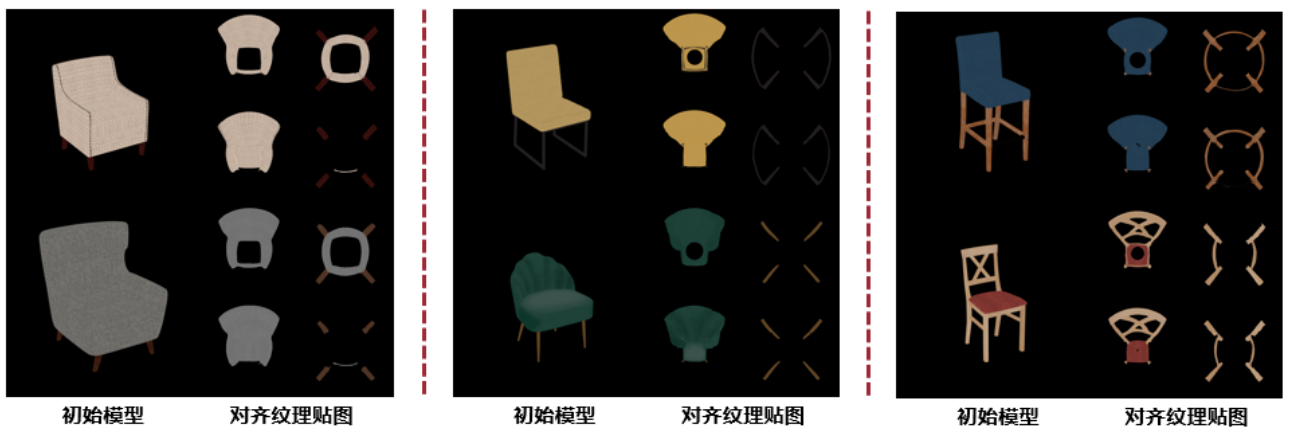


图 4. 语义对齐纹理贴图-椅子

5.3 三维模型纹理迁移结果

图 5 和图 6 展示了复现代码得到的在三维模型之间进行纹理迁移的结果，需要强调的是，基于网络重新学得的语义对齐的 UV 映射，在在三维模型之间进行纹理迁移只需要简单的替换纹理贴图即可，不需要在三维模型之间添加任何的额外操作。在图 5 和图 6 中，最左边上面展示的是初始模型的渲染图，下面展示的是对应的语义对齐的纹理贴图，右边展示的是将初始模型的纹理迁移到其他三维模型上的结果。通过观察结果可以发现，复现代码可以较好的处理三维模型之间的几何差异（如图 5 中迁移结果第一行第三个和第二行第三个，图 6 中迁移结果第一行第二个和第二行第三个），得到合理的纹理迁移结果。

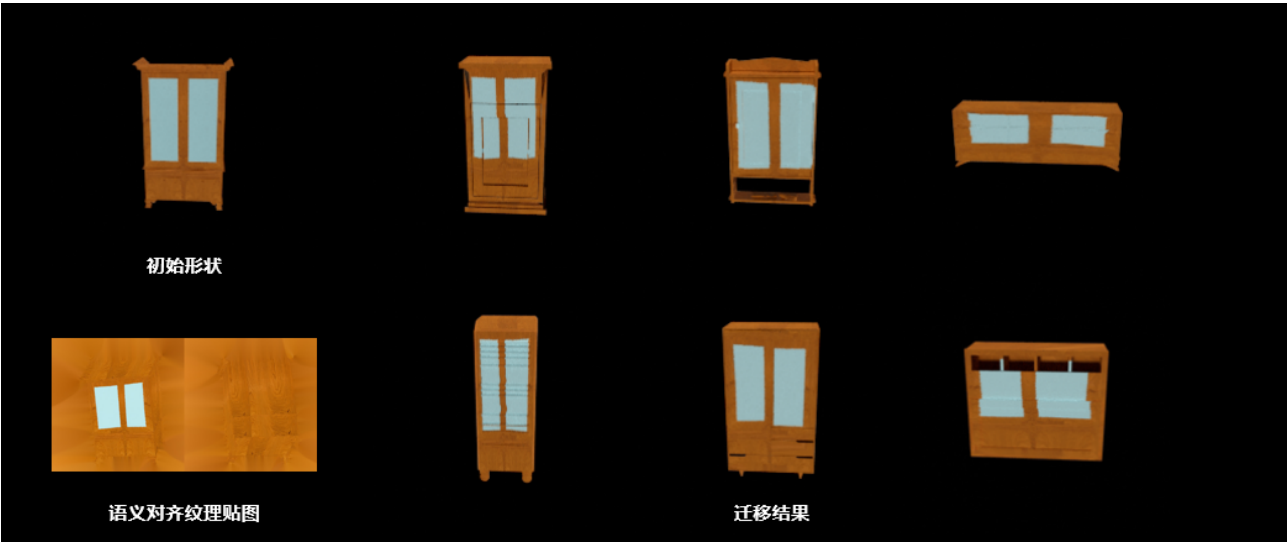


图 5. 语义对齐纹理贴图-柜子



图 6. 语义对齐纹理贴图-椅子

6 总结与展望

本项目从头复现了 AUV-Net, 在纹理迁移任务上验证结果并取得了预期效果。在对 AUV-Net 的了解和复现过程中, 我发现 AUV-Net 有以下优点: 一是使用较为简单的方法和网络结构, 通过点云重建任务, 为模型学习语义对齐的 UV 映射, 二是学习到的对齐 UV 映射降低了后续纹理迁移和纹理生成的难度。不过, 我还发现 AUV-Net 的缺点是强烈依赖人工定义的、因类别而异的先验, 而且网络需要分为多个阶段依次训练, 需要逐类别进行重新调参, 训练过程不太稳定, 我认为这样的限制可以通过引入协同分割 [7, 12] 解决。

参考文献

- [1] Anand Bhattad, Aysegul Dundar, Guilin Liu, Andrew Tao, and Bryan Catanzaro. View generalization for single image textured 3d models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6081–6090, 2021.
- [2] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.
- [3] Bindita Chaudhuri, Nikolaos Sarafianos, Linda Shapiro, and Tony Tung. Semi-supervised synthesis of high-resolution editable textures for 3d humans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7991–8000, 2021.
- [4] Wenzheng Chen, Huan Ling, Jun Gao, Edward Smith, Jaakko Lehtinen, Alec Jacobson, and Sanja Fidler. Learning to predict 3d objects with an interpolation-based differentiable renderer. *Advances in neural information processing systems*, 32, 2019.
- [5] Yu Deng, Jiaolong Yang, and Xin Tong. Deformed implicit field: Modeling 3d shapes with learned dense correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10286–10296, 2021.
- [6] Paul Henderson, Vagia Tsiminaki, and Christoph H Lampert. Leveraging 2d data to learn textured 3d mesh generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7498–7507, 2020.
- [7] Ruizhen Hu, Lubin Fan, and Ligang Liu. Co-segmentation of 3d shapes via subspace clustering. In *Computer graphics forum*, volume 31, pages 1703–1713. Wiley Online Library, 2012.
- [8] Feng Liu and Xiaoming Liu. Learning implicit functions for topology-varying dense 3d shape correspondence. *Advances in Neural Information Processing Systems*, 33:4823–4834, 2020.

- [9] Xingang Pan, Bo Dai, Ziwei Liu, Chen Change Loy, and Ping Luo. Do 2d gans know 3d shape? unsupervised 3d shape reconstruction from 2d image gans. *arXiv preprint arXiv:2011.00844*, 2020.
- [10] Shangzhe Wu, Christian Rupprecht, and Andrea Vedaldi. Unsupervised learning of probably symmetric deformable 3d objects from images in the wild. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1–10, 2020.
- [11] Kangxue Yin, Jun Gao, Maria Shugrina, Sameh Khamis, and Sanja Fidler. 3dstylenet: Creating 3d shapes with geometric and texture style variations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12456–12465, 2021.
- [12] Zisha Zhong, Yusung Kim, Leixin Zhou, Kristin Plichta, Bryan Allen, John Buatti, and Xiaodong Wu. 3d fully convolutional networks for co-segmentation of tumors on pet-ct images. In *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*, pages 228–231. IEEE, 2018.