基于多视角图像的无监督三维网格模型生成

陈祖浩
2018011114
计83
chen-zh18@mails.tsinghua.edu.cn

何雨泽
2018011351
计83
hyz18@mails.tsinghua.edu.cn

1 简介

使用平面图像、通过端到端的神经网络预测三维模型是一个新兴的研究方向，由于其相比传统三维生成步骤简单、预测耗费时间少，引起了学界广泛关注。
已有的工作包括众多单视角有监督生成三维模型、多视角有监督生成三维模型的方法，它们的效果都很好，但ground truth依赖于已有的三维模型，对数据集的要求较高。近来随着诸多可微渲染器的出现也衍生出了依赖条件最少的单视角无监督生成三维模型的工作，但由于单个图像无法反映三维模型的整体特征，因此这些网络对于复杂对象的生成效果也并不理想。
受这些工作的启发，我们希望设计一个新型神经网络，以实现无监督的、基于多视角图像的三维网格生成。多视角图像蕴含着更多物体和场景的信息，可以进一步提升生成质量；而无监督的训练方法对数据集要求小、扩展性更强，所以我们期望这样的网络今后有望向生成复杂模型与真实世界场景的方向发展。

2 相关工作

通过单视角图像、有监督训练生成三维模型。单视角图像生成三维模型是一种ill-posed问题，由于图像中蕴含的信息较少，其解不唯一，解空间也不稳定。为了解决上述问题，Abhishek Kar 提出了一种类别特化的单视角生成方法[1]，对图像进行视点估测，但此种方法无法推广至任意类别的图像；同时，由图像生成三维模型还有一个共性问题，即传统的神经网络如CNN 只能提取图像特征，而对Mesh 这种三维网格中点与点的连接关系无法加以利用，因此大部分工作转向了体素表示法三维模型生成，由于体素的特性，可以直接用CNN 对其特征进行提取。但体素表示的三维模型占用空间巨大，严重受制于目前支持的显存大小，因此用于训练的分辨率通常较低，且体素表示的三维模型给人的观感较差。Pixel2Mesh[2] 创新性的引入了GCN，可以提取三维网格中点与点间的连接关系，但其三维模型是由一个给定的椭球Mesh 逐渐上池化、形变而来的，每一步的顶点数目及其连接方式都被硬编码，灵活程度很低，对于复杂的模型，很可能出现过于平滑、顶点数不足以表达整个模型的情况。

通过多视角图像、有监督训练生成三维模型。为了弥补单视角图像信息不足、通常会在被遮挡的区域中产生粗糙几何形状的问题，通过多视角图像生成三维模型成为了一个潜在的研究方向，但多视角信息如何被有效利用仍是一个待解决的问题。Pixel2Mesh++[3] 在Pixel2Mesh 的基础上增加了多视角形变网络，它通过VGG 首先提取多视角图像的特征，再进行当前Mesh 的形变假设，最后也使用GCN 进行特征识别与形变预测。同时，该网络还允许任意次迭代输入，使生成出的三维模型越来越精细。这种基于多视角图像的形变网络非常有效，但其问题在于形变前的Mesh 是由原始的Pixel2Mesh 网络输出的，这导致了输出的三维模型同样也有Pixel2Mesh 本身具有的灵活程度低、顶点数目可能不足等缺陷。

通过单视角图像、无监督训练生成三维模型。无监督训练使用图像而不是具体的三维模型作为ground truth，这种方法的好处在于简化了数据集的要求，拓展性更强，有希望使用更大的数据集、采用更高分辨率的图像甚至实景图像构建数据集以获得更好的生成效果。但这也带来了新的问题，其中一个是loss 的计算，由于传统渲染器的光栅化步骤中丢失了深度信息，导致渲染过程是不可导的（即无法训练）。Perspective Transformer Nets[4] 使用

体素表示三维信息，并将整个空间栅格化，在投影过程中直接求max 以获得可微的loss，但不幸的是将整个空间栅格化方法仅限于体素表示法生成，且体素表示的三维模型观感较差、文件体积庞大，实际意义不大；Neural 3D Mesh Renderer[5] 使用Mesh 作为三维网格表示法，将每个像素对应的投影区域栅格化，并对三角形面片的边缘依照其所在栅格进行线性平滑处理使得渲染过程可微，但缺点在于不能对于遮挡信息进行处理，且由于每个栅格的大小一致，位于较远处的三角形面片可能被划入同一栅格中，导致信息丢失；Soft Rasterizer[6] 注重于可微渲染器的设计，同样使用Mesh 作为三维网格表示法，对于每个三角形面片都生成一张概率图而不是简单的使用栅格化进行边缘平滑处理，同时按照深度进行加权。这种方法的好处是对于被遮挡顶点与较远顶点的梯度都能有效反向传播，但其三维模型生成网络过于简单，训练过程中只使用CNN 对于图像进行特征提取，最终生成网格仅使用若干个全连接层组成的decoder 进行顶点位置的预测，忽略了点与点之间的连接关系，仍然有待改进。

我们结合上述工作，提出了一种通过**多视角图像、无监督训练**生成三维模型的方法，力图让生成出的模型更为精细、整个网络的拓展性更强。我们首先结合了Perspective Transformer Nets[4] 和Neural 3D Mesh Renderer[5] 中采用的Encoder-Decoder 生成网络进行粗糙三维网格模型的生成，然后将其送入与Pixel2Mesh++[3] 中类似的多视角形变网络进行精细化处理，这两步的渲染和Loss的计算我们都使用了Soft Rasterizer[6]中设计的渲染器完成。这种设计结合了多视角图像生成三维模型更为精细的优点，同时采用无监督训练方式，减小数据集上的限制，有望对于更复杂的物体以及现实中场景进行三维生成（现有数据集对这二者均不支持）。

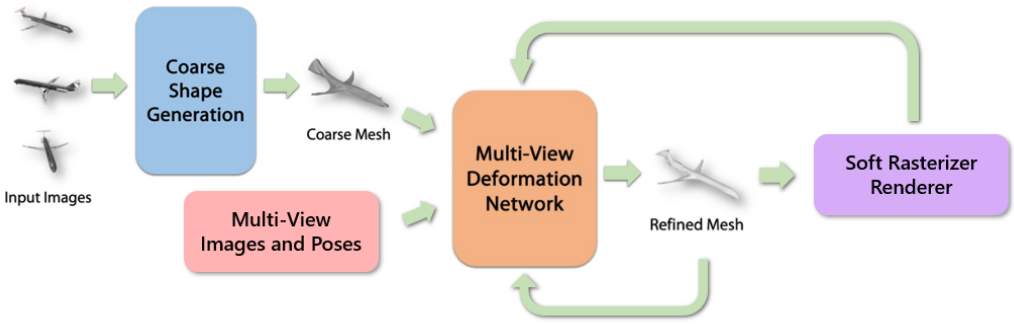


Figure 1: 本任务主要流程分为两步：第一步使用Encoder-Decoder网络生成粗糙的三维网格模型，第二步通过形变的方式对粗糙的三维模型进行精细化。两步都使用了多视角图像进行训练，且都通过可微渲染器Soft Rasterizer进行Loss的计算和梯度回传

3 方法

我们的网络主要分为两大部分：粗糙模型生成网络和多视角形变网络。由于我们设计的网络特性，在实际使用时，也可以选择仅使用单视角图像生成粗糙三维网格模型，在这之后也可以使用任意多张不同视角图像进行任意多次迭代，实现粗糙模型的精细化，每次迭代时使用的视角也可以不同。

对于第一部分粗糙模型生成，我们使用比较经典的Encoder-Decoder 生成模型。Encoder 部分我们使用Perspective Transformer Nets 中Encoder 相同的结构，由三个卷积层和三个全连接层组成，以一个RGB 图像作为输入，输出一个1x1x512 大小的潜在向量，该向量随后被送入Decoder。由于Perspective Transformer Nets 中的Decoder 目标是生成体素表示的三维模型，不适用于本项目，因此我们使用了Neural 3D Mesh Renderer 中的Decoder 结构。首先经过两个全连接层得到一个1x1x2048 的位置向量，该向量再分别通过两个全连接层生成1x1x3 的centroid 向量和1x1x3N 的bias 向量，分别代表N 个顶点相对于初始模型（通常为球或椭球）的偏移方向中心和各自的偏置，根据下列公式得到最终所有顶点的位置：

$$V_n = tanh(base_n + b_n) + tanh(c)$$

其中$base$为给定初始模型中每个顶点的三维坐标，b_n为bias向量，c为centroid向量。通过上述操作使生成的Mesh的顶点被限制在一定范围内，使得训练过程更易收敛。将变换后的顶

点信息与给定初始模型的面信息合并，即得到生成的粗糙三维模型。
之后我们会使用Soft Rasterizer渲染器渲染该三维模型的剪影图并与输入的图片对比求loss，以此来对整个网络进行迭代优化。
仅仅使用一张单视角的图像进行训练，Encoder-Decoder网络很难学习到对应物体侧面或背面的特征，所以我们对其进行了进一步的改进。我们在每次迭代的过程中会先随机选择数据集中的一个物体，然后再选择该物体随机N个视角的图像，分别送入Encoder-Decoder网络，得到N个生成的三维模型。然后我们再在这N个相机视角下对它们进行交叉渲染，最终会得到N^2张剪影图，然后以平均loss作为最终loss。这样做的目的是为了训练生成网络让其接收同一个物体的不同视角图像时能生成相同的三维模型（因为这样交叉渲染计算得到的loss值最小），这样我们就充分利用了多视角的信息使得粗糙模型生成网络在第一步就能生成出不错的三维网格模型。

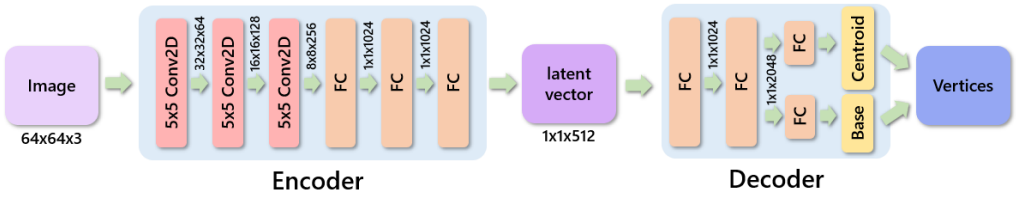


Figure 2: 粗略三维网格生成网络，输出为顶点的形变信息，对初始三维网格模型变换后得到粗糙的三维网格模型。

第二部分为多视角形变网络，通过迭代输入多视角图像及其相机参数对已有生成模型进行不断优化，具体可以分为三个步骤：第一步为形变假设采样，对于三维网格的每个顶点都假设出可能的42个形变方向，每个方向都增加一个额外的点，并把这些点与原来的顶点相连，构建局部图，用以预测Mesh顶点的形变；第二步为交叉视角特征感知，使用VGG16网络提取多视角图像的特征，将现有模型的每个顶点及其形变假设点投影到特征图中得到其二维坐标，且使用Mean、Max 等统计量信息对于多个图像的特征进行汇集，使得网络结构与输入图像数量无关，实用性更强；第三步为形变推理，使用GCN 预测每个形变方向的概率，并将其概率加权得到最终的形变量。

由于使用了GCN，相比第一步能够更有效的感知到三维网格中顶点间的连接关系，能够进行更有效的预测。同时由于网络的设计本身允许迭代，实际使用时可以多次输入多视角图像得到更好的生成结果。

具体可用以下公式表示:

$$
\begin{aligned}
I' &= \mathcal{R}(\mathcal{G}(I_1, \ldots, I_n), p') \\
G(I_1, ..., I_n) &= M_n \\
M_k &= \mathcal{D}(\mathcal{C}(f_k, \mathcal{H}_n, M_{k-1}))
\end{aligned}
$$

其中I'为生成的图像，p'为其外部参数；$\mathcal{R}$为Soft Rasterizer 渲染器，G为Encoder-Decoder生成模型；M_n为经过第n个视角迭代后产生的Mesh，f_n为第n个视角的几何特征，$\mathcal{H}_n$为第n个视角的形变假设采样，$\mathcal{C}$为跨视角特征感知器，$\mathcal{D}$为形变预测。

我们统一使用Soft Rasterizer中设计的可微渲染器对生成结果进行渲染，并与ground truth图像求loss。该渲染器的创新点在于将每个三角形面片都建立一张概率图，靠近中心的概率大、边缘的概率小；同时对于深度也进行加权，靠近相机的概率较大，最后渲染时对于每一根光线上所有相交的面片对应的概率进行归一化，其中概率图的方差是可以调整的超参数，方差越大意味着概率分布越分散、被遮挡信息的权重越大。这样做的好处在于被遮挡的顶点也可以有导数回传，同时对于栅格化等传统方法，距离相机较远的顶点结果会出现差错，而此渲染器则不会出现该问题。

我们使用的loss共分为三个部分。第一部分$\mathcal{L}_s$为剪影的交并比Loss（IoU Loss），其计算方式为$\mathcal{L}_s = 1 - \frac{\|\hat{I}_s \otimes I_s\|_1}{\|\hat{I}_s \oplus I_s - \hat{I}_s \otimes I_s\|_1}$。其中$\bigotimes$表示按元素相乘，$\bigoplus$表示按元素相加；$\hat{I}_s$和$I_s$分别表示预测的图像和真实的图像的剪影值。

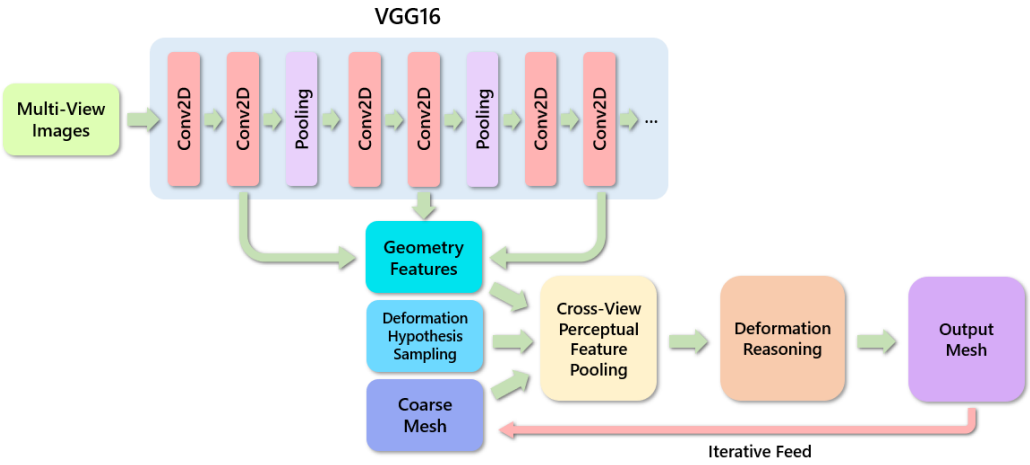


Figure 3: 多视角形变网络，通过预测可能的形变方向和大小对于粗糙三维网格模型进行精细化调整，网络本身的设计允许迭代输入以获得更好的结果。

第二部分$\mathcal{L}_{lap}$为使顶点位置分布均匀的Laplacian Loss，计算方式为$\mathcal{L}_{lap} = \sum_i ||\delta_i||_2^2$，其中$\delta_i = v_i - \frac{1}{||N(i)||}\sum_{j \in N(i)} v_j$为衡量一个点与它相邻的点集的靠近程度，$v_i = (x_i, y_i, z_i)$为顶点坐标。

第三部分$\mathcal{L}_{fl}$为平滑度的loss，使相邻的Mesh三角面片法向量更加接近，计算方式为$\mathcal{L}_{fl} = \sum_{\theta_i \in e_i}(cos\theta_i + 1)^2$，其中$\theta_i$为与边$e_i$相邻的两个三角面片的法向量夹角。

最终的Loss的计算公式为

$$\mathcal{L} = \mathcal{L}_s + \lambda\mathcal{L}_{lap} + \mu\mathcal{L}_{fl}$$

其中μ和λ均为可调节的超参数。

4 实验

我们在实验中对我们设计的两部分网络进行了分别的训练和测试，便于进行消融实验来探索各部分的实验表现。实验使用的数据集是ShapeNet数据集[1]，我们使用了其中13个类别的物体的多视角图像数据（每个物体共24个视角，每个视角仅方位角相差15度）进行训练，图像的大小是64x64像素。我们以同样使用了ShapeNet数据集的Soft Rasterizer[6]中进行的单视角无监督三维网格模型生成实验作为baseline，用交并比（Iou）的值作为实验表现的指标。

4.1 实验设置

在第一步粗糙生成模型网络的训练中，我们选择对同一个物体使用3个随机视角的图像进行交叉渲染，得到的剪影集合为

$$S = [Raa, Rba, Rca, Rab, Rbb, Rcb, Rac, Rbc, Rcc]$$

超参数方面我们使用了64的batch size和10^{-4}的learning rate。
在第二步多视角形变网络的训练中，我们同样对同一个物体选择了3个视角的图像作为特征提取的输入，求loss的时候也对应地用这三个视角对形变后的图形进行渲染，得到的剪影集合为

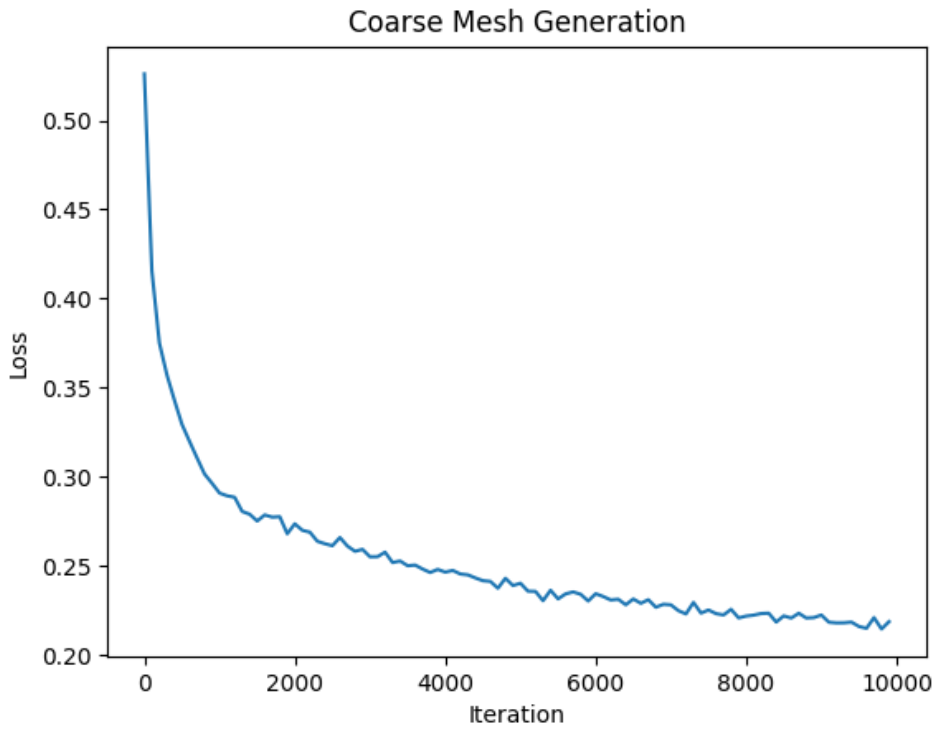
$$S = [Raa, Rba, Rca]$$

超参数方面我们使用了16的小batch size和10^{-5}的learning rate。
两步训练最终的Iteration数量分别为10000和200。

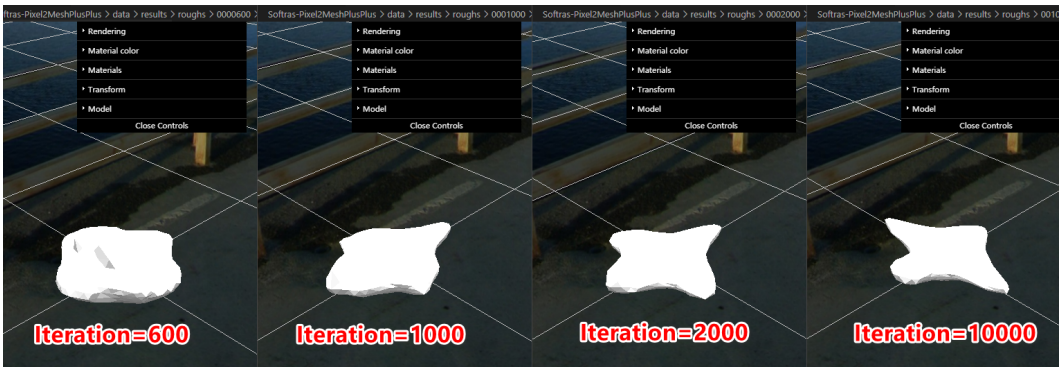
[1]https://www.shapenet.org/

4.2 实验结果

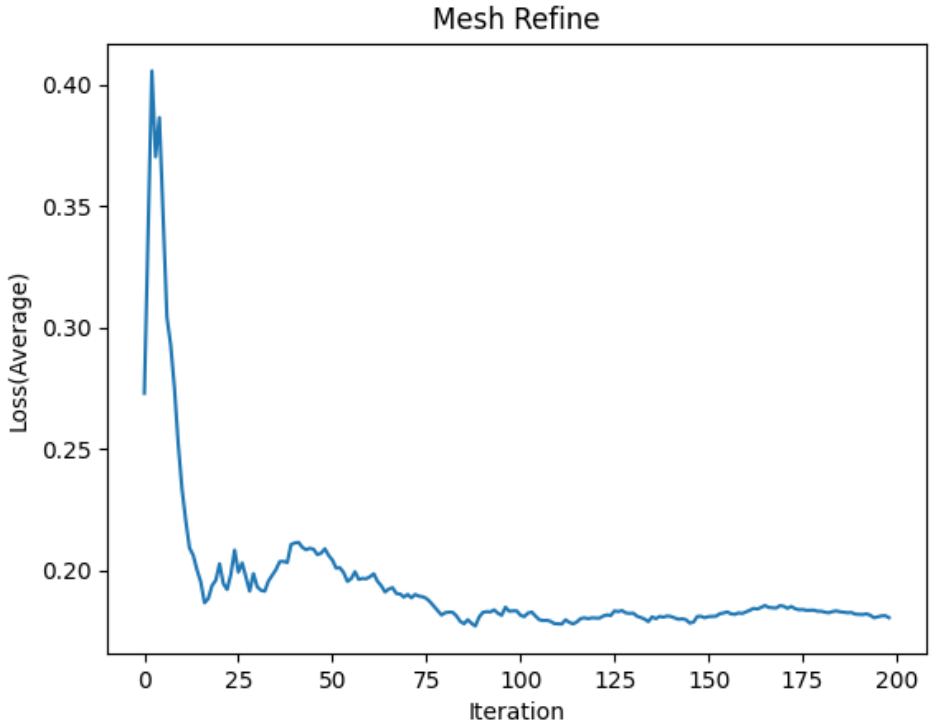
第一部分（生成粗糙模型）的训练loss曲线如下：



从loss曲线可以看出粗糙模型网络的训练效果十分好，为了更直观地观察粗糙模型网络生成的三维模型情况，我们又单独选择了ShapeNet数据集13个类中的一个类，以圆球为初始模型，重新训练了10000个Iteration，并在中间节点保存了部分物体由多视角图像生成的三维网格模型结果，如下图所示：



第二部分（模型精细化）的训练loss曲线如下：



从曲线的变化情况可以看到第二部分模型精细化的网络在前几个Iteration的时候Loss比初始的粗糙模型（第一部分的生成结果）还高，说明在刚开始参数还不确定的情况下精细化的过程并不能很好地被进行，之后则慢慢稳定在略低于粗糙模型的水平。

经过了两步训练后，我们在ShapeNet数据集上测试了13个类别的物体的生成效果，统计了它们的多视角图像通过我们的网络生成的三维网格模型重渲染得到的新图像与原始图像的交并比（IoU），并且和baseline进行了对比：

Category	Softras	Ours(without refine)	Ours(with refine)
Airplane	**0.546**	0.536	0.536
Bench	0.407	0.413	**0.416**
Cabinet	0.647	0.659	**0.660**
Car	0.691	0.700	**0.702**
Chair	0.436	**0.451**	0.449
Display	0.527	**0.546**	0.545
Lamp	0.420	0.421	**0.423**
Loudspeaker	0.612	0.623	**0.625**
Rifle	0.609	**0.613**	**0.613**
Sofa	0.607	0.620	**0.622**
Table	0.397	0.407	**0.408**
Telephone	0.713	**0.742**	0.741
Watercraft	0.551	0.558	**0.560**
Overall	0.551	0.561	**0.562**

Table 1: ShapeNet数据集上13种不同类物体的平均IoU值对比

从实验结果来看，我们的网络在第一步生成粗糙三维模型的的表现超过了我们的预期，几乎在所有类别的数据上表现都超过了baseline，但第二步的精细化的效果却并没有那么显著。
具体分析，虽然我们的第一步生成粗糙三维网格使用的Encoder-Decoder模型实际和Soft Rasterizer使用的没有太大区别，但我们在Loss的计算上使用了更多的多视角图像信息，这说明在同样的生成网络下视角数量的增加的确有利于提升三维模型生成的准确性；而我们第二步的精细化对三维模型的优化效果不显著，我们推测导致这个结果的主要原因是我们在第二步精细阶段使用的多视角形变网络对较大的batch size支持性很差，训练的性能随着batch size的增大会急剧降低，同时还会出现显存不足的情况，这使得我们不得不使用和第一步严重不匹配的小batch size进行训练（即便如此每一个Iteration的运行时间还是很长），因此训练的收敛速度和效果都受到了很大的影响。

5 总结

我们构建了一个通过多视角图片、以无监督方式训练、端到端生成三维网格模型的神经网络，提供了新的研究方向，并取得了比之前相关工作更优的结果。
但我们的工作也有许多不足之处，最主要的不足之处就在于没有预估好第二部分精细化模型网络训练的性能开销，导致最终我们最后不能使用理想的超参数迭代足够长的次数。
今后我们会进一步尝试网络结构的调整与优化，降低显存的占用，并尝试构建分辨率更大的虚拟模型数据集、真实物体数据集进行训练以检验网络的可扩展性。

参考文献

References

[1] Kar, A., S. Tulsiani, J. Carreira, et al. Category-specific object reconstruction from a single image, 2015.

[2] Wang, N., Y. Zhang, Z. Li, et al. Pixel2mesh: Generating 3d mesh models from single rgb images, 2018.

[3] Wen, C., Y. Zhang, Z. Li, et al. Pixel2mesh++: Multi-view 3d mesh generation via deformation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1042–1051. 2019.

[4] Yan, X., J. Yang, E. Yumer, et al. Perspective transformer nets: Learning single-view 3d object reconstruction without 3d supervision. *arXiv preprint arXiv:1612.00814*, 2016.

[5] Kato, H., Y. Ushiku, T. Harada. Neural 3d mesh renderer, 2017.

[6] Liu, S., T. Li, W. Chen, et al. Soft rasterizer: A differentiable renderer for image-based 3d reasoning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7708–7717. 2019.