硕士学位论文

基于深度学习的三维重建算法研究

RESEARCH ON 3D RECONSTRUCTION ALGORITHM BASED ON DEEP LEARNING

郭帅君

哈尔滨工业大学

2021 年 6 月

国内图书分类号：O244　　学校代码：10213
国际图书分类号：519.6　　密级：公开

理学硕士学位论文

基于深度学习的三维重建算法研究

硕士研究生：郭帅君
导　　师：石振锋副教授
申请学位：理学硕士
学　　科：计算数学
所在单位：数学学院
答辩日期：2021 年 6 月
授予学位单位：哈尔滨工业大学

Classified Index: O244
U.D.C: 519.6

Dissertation for the Master Degree in Science

RESEARCH ON 3D RECONSTRUCTION ALGORITHM BASED ON DEEP LEARNING

Candidate: Guo Shuaijun
Supervisor: Associate Prof. Shi Zhenfeng
Academic Degree Applied for: Master of Science
Speciality: Computational Mathematics
Affiliation: School of Mathematics
Date of Defence: June, 2021
Degree-Conferring-Institution: Harbin Institute of Technology

摘 要

三维重建是计算机视觉领域中的一个重要课题，三维模型有助于人们更好地去认知客观世界中的物体。从二维的 RGB 图像恢复物体的三维结构是一个典型的反问题，因此需要结合图像的一些先验知识完成重建。近些年来深度学习因为其强大的表达能力和学习能力，在图像相关领域取得了巨大的成果，可以利用它来推断物体的三维结构。三维模型在计算机中有多种表达形式，它们各具特点，适用于不同的应用场景当中，其中比较具有代表性的是体素以及网格模型。因此，本文从神经网络的角度出发，分别设计了新的重建体素模型以及重建网格模型。

对于三维重建体素任务，本文模型首先使用基于视觉 transformer 的图像编码层提取图像的结构性特征，接着利用基于三维转置卷积的解码层将特征信息转义到体素空间上，使用基于三维视觉 transformer 模块和三维卷积模块的输出层得到体素概率值，最终输出分辨率大小为 $32\times32\times32$ 的体素模型。对于多视图任务，本文设计了一种基于三维卷积的注意力模块去融合不同图像输出的体素结果，不同于许多基于循环神经网络的融合模块，该方法能够效率的利用不同视角的图像完成重构。通过实验表明，本文方法能够在单视图和多视图的情况下快速的恢复出物体的体素。该体素模型能够很好的反映原物体的整体结构。

对于三维重建网格任务，考虑到物体结构的多样性，本文模型首先利用卷积层生成一个粗略的体素模型，将该体素模型利用立方化的方法转换成三角网格模型，并将提取到的二维特征以及三维特征投影到网格中的顶点，接着利用带有残差连接和恒等映射的多层图卷积网络变形三角网格。结果说明了本文方法能够利用单张图像恢复出具有物体纹理、线条等细节的三角网格模型。

关键词：三维重建；深度学习；体素；图卷积神经网络；三角网格模型

Abstract

3D reconstruction is an important topic in the field of computer vision. 3D model can help people to better understand the objects in the objective world. However, restoring the three-dimensional structure of an object from a two-dimensional RGB image is a typical inverse problem, which needs the combination of some prior knowledge of the image. In recent years, due to the powerful expression and learning capabilities, deep learning has achieved great results in image-related fields. In addition, it can be used to infer the three-dimensional structure of objects. There are many kinds of expression forms of 3D model in computer, they have their own characteristics and are suitable for different application scenarios. Among them, the representative ones are voxel and mesh model. Therefore, from the perspective of neural network, this paper designs a new voxel reconstruction model and mesh reconstruction model respectively.

For the task of voxel reconstruction, this model first uses the image encoder based on vision transformer to extract the structural features of the image, then uses the decoding layer based on 3D transposed convolutional layers to escape the feature information to the voxel space. Finally, the output layer based on 3D vision transformer module and 3D convolutional module is designed to obtain the probability value of the voxel and output the voxel model with the resolution size of $32\times32\times32$. For multi-view tasks, this paper designs an attention module based on three-dimensional convolutional layers to fuse the voxel results of different image outputs. Unlike many fusion modules based on recurrent neural networks, this method can efficiently use images from different perspectives to complete reconstruction. The experiments show that the method in this paper can quickly recover the voxel of the object in single view and multi-view situations. The voxel can well reflect the whole structure of the original object.

For the task of 3D mesh reconstruction, with the consideration of the diversity of object structures, this model first uses convolutional layers to generate a coarse 3D voxel model and then transforms the 3D voxel model into a triangular mesh model by using cubify method. Next, this model project the extracted two-dimensional features and three-dimensional features to the vertices of the mesh,

then use the multi-layer graph convolutional neural network with residual connection and identity mapping to deform the triangular mesh. The results show that the proposed method can use a single image to restore the triangular mesh model with object texture, lines and other details.

Keywords：3D reconstruction, Deep learning, Voxel, Graph convolutional neural network, Triangular mesh

目　录

第 1 章　绪　论

1.1　课题研究背景与意义

近些年来，随着科学技术的飞速发展，计算机视觉技术也相应的发展了起来。通过模拟人们的视觉系统，计算机视觉可以对现实世界之中的物体形成认知。计算机视觉通过对二维图像像素值的处理和分析，能够推断出图像的更深层次信息，比如说图像中物体的轮廓，图像中物体的所属类别，或者不同图像之间的关联点，由此，计算机视觉有了许多令人印象深刻的应用，比如说图像去模糊[1,2]、图像分割[3]以及逐渐成熟的人脸识别技术[4]，这些应用都解决了许多实际问题。

人们在客观世界之中接触到的物体是三维的，我们可以从立体化角度的去观察物体，这样就可以更好的分析它们的结构和性质，比如说汽车的车内空间是否足够宽阔以便于人们可以舒适的坐进去；书包是否有夹层，学生们是否可以利用多个夹层来归纳整理不同的书籍。但对于计算机视觉来说，这样的解析过程是很有难度的，因为物体在计算机中的一般表现形式是二维图像，这一表达形式相对于三维物体来说会有很多的信息损失，所以在一些应用中需要通过一定的技术手段来恢复出物体的三维结构。三维重建技术正是基于这一想法，通过利用物体中的某些局部信息，比如说利用 RGB 图像或者深度图恢复出它的三维结构。

三维重建也有着重要的实际意义，例如在医学医疗之中，可以将医疗图像重建成对应的三维结构[5]，便于医生诊断信息；也可以在游戏当中引入虚拟现实[6]技术，通过对人体的三维建模，模拟出真实的运动情况；在古文物研究中，利用三维数字修复技术可以帮助研究人员恢复古建筑、古瓷器的三维形状。这些应用都说明了三维重建的现实意义。

三维物体在计算机中有不同的表现形式。现阶段大致可以分为四类：深度图、体素、点云以及网格。它们都分别具有不同的特点，深度图是一张二维图像，其中的每个像素点代表着视点到遮挡物表面的距离，可以显示出物体的深度域，在计算机中占用存储空间较少，缺点是图像的表面轮廓显示的比较粗糙；体素是体积像素的简称，三维坐标中每个点的 0 或 1 代表其是否含有体素块，这种表现形式的优点是计算机容易处理，缺点是当分辨率变高时，其所占用空间也会成几何倍数的增长；点云是三维物体中点的数据集合，其包含了物体的

大量信息，包括颜色信息、坐标信息，数据量较大，但是其没有考虑到物体点与点之间的关联；网格包含三维物体的坐标信息、边信息、面信息，这种表现形式更容易看出物体的结构，但是计算机处理起来难度较大。如图 1-1 分别显示了深度图、体素例子。可以看到，物体的深度图显示比较模糊。

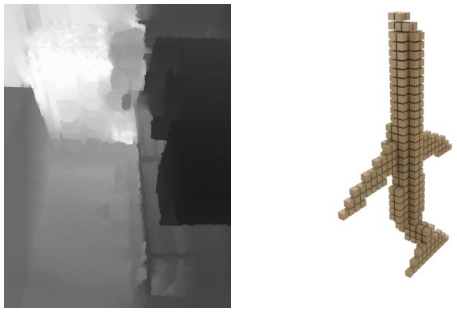

图 1-1　物体深度图[7]、体素例子

而从二维图像恢复物体的三维结构本质上是一个反问题，现阶段许多方法利用图像序列的特征点，结合图像序列之间的联系，实现三维重构。但是，由于一些关键信息的缺失，会给重建的过程带来一些困难：如何对于不可见部分进行还原就是在三维重构过程中必须考虑的问题，这个问题的解决需要利用特定技术来对图像的三维结构进行推测；当多幅图像作为输入时，如何将不同角度的图像关联起来，在面对纹理较弱的图像序列时，这个问题会变得更加严峻；有些方法需要将相机提前标定，因此在特定场景之下并不适用；当物体表面的一部分被自遮挡时，图像的特征点可能无法提取，导致重建的失败。因此，三维重建这一课题还有许多需要解决的问题。

近些年来，随着理论的进步以及计算机算力的发展，神经网络因为其强大的表征能力受到了越来越广泛的关注。具体到计算机视觉，通过卷积层的学习，神经网络可以学习图像不同层级的几何特征以及结构特征，接着通过提取这些特征，可以应用到图像识别、图像分割、图像去噪等等领域。神经网络也具有较好的推测能力，可以考虑利用这一特性，使用物体中的可见的部分对不可见部分进行推测。

三维模型中存在大量数据是不规则的，比如说三维网格中的顶点、边等等的数据结构，普通的卷积层在处理规则的图像数据时表现的很好，但是对于非固定结构的数据却无能为力。图卷积网络[8,9]可以解决这一问题，该网络利用图中节点的特征，以及各个节点之间的关联性去得到下一层的特征矩阵，而如果将三维网格中的顶点看作是图的节点，传统卷积层提取到的特征看作是节点的特征，顶点之间的边看作是节点之间的关联，三维网格这一模型就可以应用到图卷积网络中。因此可以使用图卷积网络去处理三维网格模型。

多视图重建任务相对于单视图重建任务来讲，能够更全面的获得物体的信息，因而能够重构出效果更好的三维模型。但是，多视图重建任务必须考虑的一个问题是如何效率的利用图像之间的关联性，普通的卷积神经网络对单个图像的特征提取是很有效的，但是并没有融合多幅图像之间特征的能力。如果使用循环神经网络序列的提取不同角度的图像特征，会使模型的复杂度变高。基于上述考虑，模型应当具备平行的提取图像特征，并且将单视图重建结果效率融合的能力。这样设计模型的好处在于不需要额外的参数去学习多视角的图像特征，同时也能够平行的利用不同角度图像去重建三维结构，输出结果与图像的输入顺序无关。

综上所述，本文考虑使用深度学习的方法，分别完成物体的重建体素任务以及重建三角网格任务。

1.2 国内外研究现状

三维重建的目标是获得物体在计算机中的三维模型表示，现阶段的方法可以分为两大类：一种是通过三维扫描仪等仪器扫描物体获得三维模型，称为主动方法；另一种是利用相机所获得的物体图像，经过图像配准、立体几何等等方法获得物体的三维表示，称为被动式的三维重建方法。三维扫描仪方法的精度较高，但同时，该类方法所采用的仪器较为精密，要求较高，也可能对部分对象如文物、毛发等等造成侵略性，适用场景较少。

从图像恢复物体的三维结构是一种典型的反问题，因此传统的三维重构方法利用一些特定的图像先验知识去完成重建。

纹理形状恢复法[10](Shape From Texture)根据物体中同一纹理单元在不同角度的纹理特征的形状变化，测算出对应的深度信息。该方法产生的三维结构比较粗糙，而且对目标物体的纹理特征要求较高，需要目标的纹理信息比较细致，因此有一定的局限性。

阴影形状恢复法[11](Shape From Shading)是基于光源照射物体所产生的阴影而完成三维重建任务的。该方法利用不同光照条件之下物体的明暗程度建立方程，通过亮度约束、光滑性约束、单位法向量约束等求解方程的解，最后再通过一些数值方法求出物体的三维模型，该方法的应用较为广泛，但缺点是对光源的要求比较高，需要特定条件的光源位置、光照强度、光照方向，也无法处理复杂光源的情况。

从包含运动信息的图片序列中恢复三维结构的技术近些年来发展迅速，其中比较有代表性的工作是 SFM[12](Structure from Motion). SFM 首先利用

SIFT[13]、SURF[14]等算法提取序列图像之间的特征点，特征点的选取会考虑到图像尺度、旋转等因素；接着通过特征匹配将图像进行两两匹配，这一步可以使用粗暴匹配法和临近搜索法，然后利用采样一致性算法对相似点计算基础矩阵；第三步利用初始化像对计算出单应矩阵；第四步首先计算本征矩阵，通过本征矩阵的 SVD 分解得到第二个图像的旋转平移矩阵，再通过矫正畸变可以得到点的空间位置；第五步是将后续图像加入进来，通过之前步骤计算的点的空间位置以及特征点的对应关系，可以计算出后续图片的 R, T 矩阵；第六步是利用光束平差法优化误差。SFM 方法的优点是重建效果精度较好，速度也较快，但是现阶段该方法仍然存在着两个问题：第一，当视点跨越一个较大的范围之后，特征点的匹配将会变得困难；第二，在某些实际情况之中无法得到物体表面的完整信息，这就会使得物体重建不完整，有一定的局限性。

SFM 算法重建的结果是稀疏点云，原因在于重建的点一般为特征点，特征点是稀疏的。MVS[15]算法则可以生成稠密化的点云，该方法利用相机模型参数已知的图像进行像素匹配，首先建立一个一致性判定函数，接着在后续的迭代过程中对参数进行传播。图 1-2 展示了传统方法三维重建的一般流程。

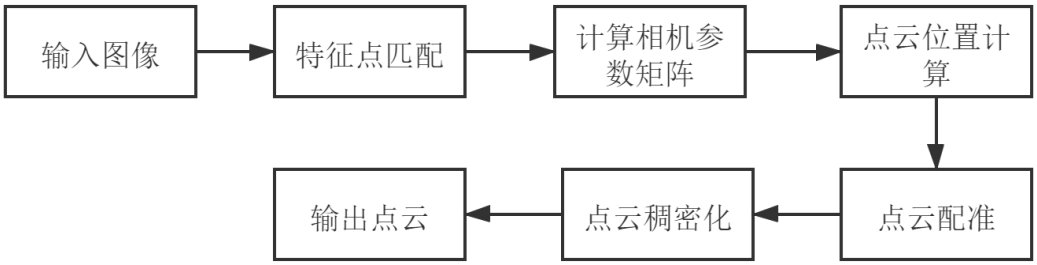

图 1-2　传统方法三维重建一般流程

随着 RGB-D 相机的出现，基于 RGB-D 图像的三维重建技术也有了广泛的发展。RGB-D 相机不仅仅可以获得物体的色彩位置信息，同时也可以获得到物体的深度信息。Newconbe[16]等人提出了一种利用深度相机重建室内场景的实时方法，该方法首先构建全局隐式表面模型(TSDF), TSDF 首先初始化一个长方形的包围盒，然后根据分辨率的要求均匀划分网格节点，每个节点的数值代表着其与最近物体表面的距离，数值为正则认为它在物体表面的前方，数值为负则代表它在物体表面的后方，因此，通过寻找等值面就可以找到物体的表面。接下来，该方法将 Kinect 相机采集到的深度图片实时的融入到 TSDF 模型当中去更新数值，通过迭代最近点算法进行位姿估计，最终得到了三维模型。该方法使用稠密化的表示方式，算法的内存占用较高，同时在重建较大物体时累计误差会造成重影现象，因此该方法的应用场景为室内，无法重建较大物体。许多

学者在此基础上进一步的提出了新的重构模型，Thomas[17]等人利用表面单元表示方法完成重构，该方法将小规模的局部闭环和大规模的全局闭环结合在一起，保证了重构的全局一致性；Angela[18]等人使用 SIFT 方法提取序列图像间的特征关系，利用体素哈希算法重建模型，实时估计 BA 优化姿态，得到了较好的重建效果。基于 RGB-D 的三维重构算法优点是重建的精度较高，在某些场景之下可以做到实时预测并不断优化结果，但是缺点是当模型需要处理的数据量较大时，由于内存不够会使重建变得困难，在采集图像的时候也需要特殊的 RBG-D 相机，对设备有一定的要求。

近些年来，深度学习方法由于其强大的数据处理能力受到了广泛的关注，许多的研究人员从神经网络的角度出发去研究三维重建，希望利用神经网络提取图像的深层次特征，并利用这些特征完成重建，这一方向在近些年来受到了广泛的关注。

Choy[19]等人于 2016 年使用深度学习的方法重建体素。该方法使用卷积层编码图像并利用特征学习体素概率，对于多视图重建的问题，提出了三维卷积 LSTM 网络和三维卷积 GRU 网络处理图像序列数据，是三维重建领域第一篇使用深度学习方法能得到较好结果的方法，该方法的优点是能够处理序列图像数据，根据不同视角的图片完善三维重构的结果，但是仍然有以下几点的不足：1.由于 LSTM 方法的长时记忆损失，在三维重建的过程中图像的特征不能很好利用，重建结果细节缺失较多 2.该方法处理的是序列数据，因此对于不同顺序相同内容的序列图片，重建的结果会有差异[20]3.处理类似结构的数据时计算不并行，该模型的参数数量较多。如图 1-3 展示了三维 LSTM 网络和三维 GRU 网络的计算过程。

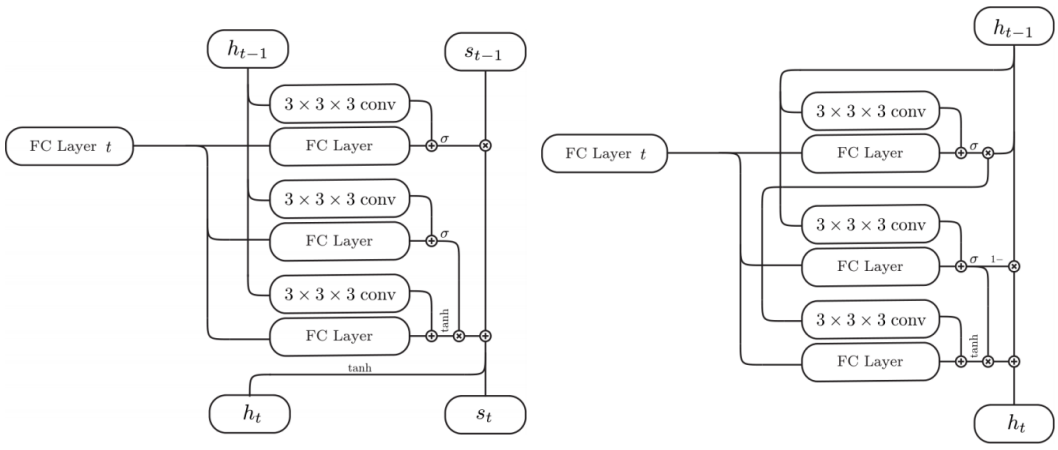

图 1-3　三维 LSTM 网络和三维 GRU 网络[20]

考虑到 RNN 在处理序列图像数据时的不足，Wang[21]等人于 2017 年提出

了 Denseinet, 该方法使用最大值池化的方法去聚合不同角度图像特征，Paschalidou[22]等人提出的 Raynet 使用平均池化的操作融合多角度的特征，这类基于池化的方法一定程度上解决了多角度图像特征融合的问题，但是池化的操作会带来一定的信息损失，影响最后的重建结果。

考虑到池化操作会带来的信息损失，Xie[23]等人提出 Pix2vox 模型，该方法使用二维卷积层提取图像特征，接着使用一种上下文融合感知模块融合不同角度图像重建的体素，最后使用三维卷积层输出结果。该方法的优点是提取图像特征的参数是共享的，并且能够根据图像的信息学习图像融合层的权重，在重建速度和质量上都有提升。

上述介绍的几种方法是根据 RGB 图像重建体素的，并且一般来讲不需要图像对应的相机参数，重建体素的优点是易于计算机处理，数据是典型的欧式空间数据，模型重建的速度也较快，但是同时这也会带来一些缺点：1.现阶段各种方法重建体素的分辨率大多为 32^3 或者 48^3，重建的精度不够，但是随着分辨率的提升，计算复杂性也会成几何倍数的提升，重建的难度也会变大 2.体素模型缺乏细节以及纹理特征，重建的模型可视化效果一般。因此也有一些研究人员尝试从点云的角度重建三维物体模型。

PSGN[24]方法使用单幅图像恢复点云，该方法将图像作为输入，接着通过卷积层学习特征，使用沙漏网络结合局部与全局特征，由于二维到三维的不确定性，添加多个不同的扰动项进行多次预测。并且，该方法中利用了两种新的损失函数用于三维重建：Chamfer 损失函数和 EMD(Earth Mover’s distance)损失函数，能够约束目标点云与重建点云之间的距离。

Wei[25]等人于 2019 年提出一种可以对不可见部分建模的方法，对于输入图像，加入多个噪声得出不同的重建结果，再取交集得到结果，在测试部分，使用了一致性损失来进行在线优化。该方法重建结果和 PSGN 相比也有了一些提升。

对比基于体素的重建模型来说，基于点云的重建结果更清晰，灵活性更强，并且不会受到分辨率的限制，但是该类方法仍然没有考虑到三维模型中点与点之间的关联，对模型的纹理描绘不够精确，可视化效果不好。

2018 年 Wang[26]等人提出了一种使用图卷积网络重建三维网格的方法。该方法以图像和初始化的网格球体作为输入，使用 VGG 网络提取图像特征，利用一种具有残差结构的图卷积神经网络变形网格，并且使用图的反池化方法增加网格顶点数目，最终得到精细化之后的三角网格结果。该方法考虑了三维网格中点与点之间的联系。

2019 年 Georgia[27]等人提出 Mesh R-CNN 网络将 R-CNN[28]与三维重建联系起来，首先使用 R-CNN 检测目标所在位置，接下来使用图卷积网络重建目标网格，该方法对现实场景中的物体重建效果较好，并且能够重构复杂物体的网格结构。

基于网格的三维重建结果可视化的效果较好，同时重建的算法也会比较复杂，不但要考虑点坐标这一非欧式数据，同时也要考虑网格中边和面的存在，模型会变的比较复杂。

2019 年 Lars[29]等人尝试使用一种新的三维表示方法完成三维重建，该方法提出一个占用方程来预测空间中的点是否在三维模型内部，接着利用多分辨率等值面提取方法形成三维表面。该方法可以预测任意分辨率模型，这种表示方法令人耳目一新。

总的来说，主动的三维重建方法效果最好，它能直接对物体扫描得到物体精确的三维结构，但是不适用于一些现实场景，局限性较大。传统的三维重建方法在特定条件之下可以完成效果较好的重构结果，该类方法更多的利用图像的先验知识以及图像之间的关联性完成重构，一般来讲对于图像的要求比较高。基于 RDG-D 图像的三维重建方法具有重建速度快的优点，但是需要深度相机的支持，对于设备的要求较高，现阶段也存在当场景较大时重建失败的情况。基于深度学习的三维重建方法优点在于其对于图像的要求较少，甚至有些方法可以完成基于单张图像的重建，但是其依赖于大型数据集的构建。

1.3 本文主要研究内容以及章节安排

本文的主要研究内容是基于深度学习去重建三维物体模型。主要分为两个部分，首先是利用 RGB 图像恢复出物体的体素模型，进一步的，利用图卷积网络完成物体的三角网格模型重建。本论文的章节安排如下：

第一章，绪论：首先对三维重建的研究背景进行了阐述，同时介绍了计算机中三维物体不同的表示方法。接下来对研究现状进行了细致的分析，从传统的三维重建方法开始介绍，接着介绍了基于深度学习的重建方法、基于 RGB 相机的重建方法。对比了这些方法的优缺点，给出了本文研究方向以及论文的章节安排。

第二章，相关理论与技术：首先介绍了物体从三维空间投影到二维像素坐标系的过程，讲解了相机的畸变公式，接着说明了全连接神经网络和卷积神经网络的计算过程，分析了两种神经网络的特点，并说明了它们的应用场景。最后对两种经典的卷积神经网络模型 ResNet 和 U-net 模型分别进行了介绍，类比

了它们之中的关键结构。

第三章，三维重建物体体素：首先对三维重建体素这一课题进行了分析，接着设计了一种编码器-解码器的网络结构完成单视图重建任务，该模型利用基于视觉 transformer 的编码层提取图像特征，并且设计了一种三维视觉 transformer 模块去输出物体体素概率值。对于多视图的情况，利用一种基于三维卷积的注意力模块完成重建任务，最后对结果进行了分析以及对比。

第四章，三维重建物体网格：首先分析了三维网格模型的特点，设计了一种利用图像不同尺度特征来变形网格结构的模型，该模型由卷积神经网络和一种带有残差连接的图卷积神经网络构成。通过实验分析了模型的有效性，同时对网络的损失函数进行了讨论，给出了两种结果，一种是可视化效果较好但评估指标较差的情况，另一种是可视化情况较差但评估指标较好的情况。

最后总结了本文的主要内容，并对未来的研究方向进行了简要的展望。

第 2 章　相关理论与技术

2.1　相机成像模型

三维重建的目标是通过二维图像还原出物体的三维表示，研究这一课题的前提是了解摄像机如何将现实世界中的物体投影到图像空间当中去的。这个过程涉及到四种坐标系之间的变换：世界坐标系、相机坐标系、图像坐标系、像素坐标系。接下来本节将介绍这几个坐标系之间的关系。

世界坐标系的意义为三维现实世界中的坐标系，它以客观世界中的某个点作为原点，而相机坐标系则是以相机的光心作为原点。具体的，以齐次坐标的形式表示世界坐标系下的点：$(x_w, y_w, z_w, 1)$ 和相机坐标系下的点：$(x_c, y_c, z_c, 1)$，两个坐标系点之间的变换涉及到旋转及平移. 首先来看旋转变换，假设坐标系围绕 z 轴逆时针旋转，旋转角为 θ，可以得到两个点之间的转换公式为：

$$x_c = x_w \cos\theta + y_w \sin\theta$$

$$y_c = y_w \cos\theta - x_w \sin\theta$$

$$z_c = z_w$$

写成矩阵形式就可以表示为：

$$\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} = R_z \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix}$$

R_z 代表绕 z 轴的旋转矩阵，类似的，绕 x 轴以及绕 y 轴的旋转矩阵可以表示为：

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\omega & \sin\omega \\ 0 & -\sin\omega & \cos\omega \end{bmatrix},\ R_y = \begin{bmatrix} \cos\varphi & 0 & -\sin\phi \\ 0 & 1 & 0 \\ \sin\varphi & 0 & \cos\phi \end{bmatrix}$$

其中，ω, φ 分别代表绕 x 轴和绕 y 轴的旋转角。这样，旋转矩阵就可以表示为 $R = R_x R_y R_z$. 而平移的过程就是在旋转的基础上添加常量，表示为：

$$\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = R \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}$$

其中 t_x, t_y, t_z 分别代表 x 轴，y 轴，z 轴上的平移量。令：

$$T=\begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}$$

则上述两个坐标系的变换公式就可以表示为：

$$\begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix}=\begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix}\begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}$$

该变换是欧式变换，因此可以保证物体两点之间的距离不变、两点之间的夹角不变，物体变换前后的体积不变。

接下来可以利用小孔成像的原理得到相机坐标系上的点到图像坐标系上点的变换公式。如图 2-1 展示了相机坐标系的物体到图像平面的映射。其中(x,y)代表对应点在图像坐标系下的坐标，f代表相机的主点到焦点的距离，也就是相机的焦距。通过图上的几何关系可知：

$$\frac{x_c}{x}=\frac{y_c}{y}=\frac{z_c}{f}$$

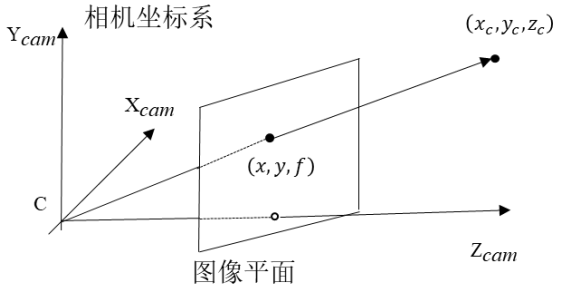

图 2-1　相机坐标系到图像坐标系的转换

由此关系就可以得到图像上点的坐标位置：

$$x=\frac{x_c}{z_c}f, y=\frac{y_c}{z_c}f$$

可以将上述的转换关系用矩阵表示为：

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}=\frac{1}{z_c}\begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}\begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix}$$

最后两种坐标系之间的区别是坐标系原点的位置和单位：图像坐标系的原点是图像的中心，单位是米；像素坐标系的原点是图像的左上角，是以像素作为单

位的。图 2-2 直观的表示了两者之间的差别。

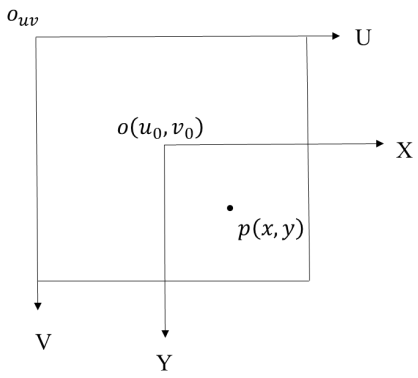

图 2-2　图像坐标系与像素坐标系的关系

设 dx 和 dy 分别代表每个像素点在图像坐标系中的长度和宽度，(u_0, v_0)代表图像中心点在像素坐标系中的位置。则两者之间的转换关系可以写为：

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{dx} & 0 & u_0 \\ 0 & \frac{1}{dy} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

这样，将上述三个步骤结合起来，相机模型将世界坐标系下的点映射到像素坐标系下的转换公式就为：

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \frac{1}{z_c} \begin{bmatrix} \frac{1}{dx} & 0 & u_0 \\ 0 & \frac{1}{dy} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}$$

而现实中相机将物体投影到图像平面时会产生畸变。畸变主要有两种类型：径向畸变、切向畸变。径向畸变产生的原因是光线距离透镜中心越远，弯曲的越严重。矫正公式为：

$$x_l = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6)$$

$$y_l = y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6)$$

其中(x_l, y_l)代表图像坐标系经过矫正之后点的对应位置，$r^2 = x^2 + y^2$．k_1, k_2, k_3代表相机径向畸变参数。

产生切向畸变的原因是成像透镜与成像平面不平行。矫正公式可以写为：

$$x_r = x + \left[2p_1xy + p_2\left(r^2 + 2x^2\right) \right]$$

$$y_r = y + \left[p_1\left(r^2 + 2y^2\right) + 2p_2xy \right]$$

p_1，p_2代表相机切向畸变的参数。在获得相机畸变参数之后，就可以通过上述的矫正公式，获得矫正畸变之后的图像点。

本节介绍了相机是如何将三维物体中的点映射到像素坐标系当中去的，通过上述的讨论可知，从三维物体到二维像素点的过程当中丢失了许多信息，因此三维重建这一问题是不适定的，需要考虑应用一些图像的先验知识结合相关理论完成三维重建任务。

2.2 神经网络知识介绍

人们可以通过视觉完成各种各样复杂的任务，而这依赖于人们将视觉得到的信息反馈到神经系统中处理，神经系统再通过各种各样的神经元完成信息的交换、传递、处理。而神经网络通过感知器模拟神经元去完成信息的处理过程。随着近些年来的快速发展，神经网络在计算机视觉、自然语言处理[30,31]等方向展现出了巨大的潜力。

2.2.1 全连接神经网络

神经网络是由多层感知器构成的。一个简单的线性感知器的结构如图 2-3 所示。感知器由输入权值、激活函数组成。

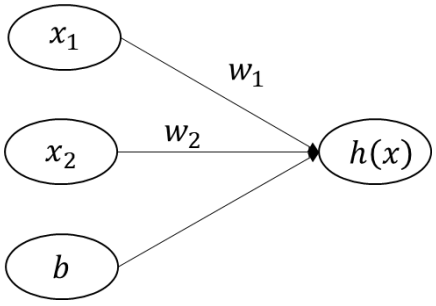

图 2-3 线性感知器结构

设激活函数为f，则该感知器的输出为：

$$h(x) = f\left(w^T x + b\right), w = \left[w_1, w_2\right], x = \left[x_1, x_2\right]$$

其中w代表权重，x代表输入，b代表偏置项。激活函数是为了在神经网络中引入非线性性质，从而能更好的处理各种复杂的数据。Sigmoid 激活函数的公式为：

$$S(x)=\frac{1}{1+e^{-x}}$$

它能将输入映射到$(0,1)$的区间之上，并且它的梯度计算公式相对简单，另一种常用的激活函数是 tanh 函数：

$$T(x)=\frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$$

神经网络就是按照一定规则将这些感知器连接起来的。图 2-4 显示了包含一层隐藏层单元的全连接神经网络。该全连接神经网络的每层输出就可以表示为：

$$x^{(2)}=f\left(w^{(1)}x^{(1)}+b^{(1)}\right)$$

$$y=f\left(w^{(2)}x^{(2)}+b^{(2)}\right)$$

其中$w^{(i)},x^{(i)},b^{(i)},i=1,2$分别代表第$i$层的权重、输入以及偏置，$y$代表该全连接神经网络的输出。这是神经网络前向的计算过程。可以看到神经网络的表现很大程度上的依赖于权重的数值，因此需要利用反向传播的方法更新网络之中的参数，也就是训练网络的过程。一种常用的方法是利用梯度下降算法更新模型中的参数：

$$w_{ij}=w_{ij}-\eta\frac{\partial E}{\partial w_{ij}}$$

其中，E代表损失函数，含义是真实值与预测值之间的差距，w_{ij}代表神经网络中第i层的第j个权值，η代表学习率。最后，利用这些训练好的参数在测试集上评估结果。

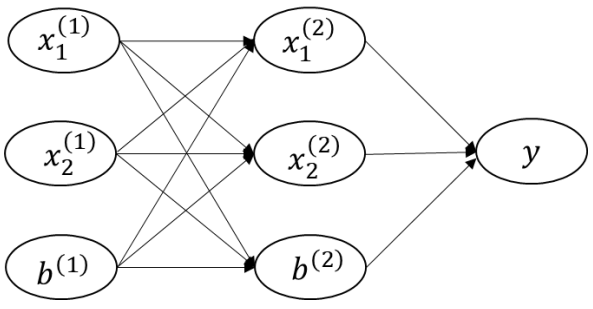

图 2-4　全连接神经网络例子

2.2.2　卷积神经网络

全连接神经网络在许多任务中取得了不错的效果，但是在处理图像数据时，它存在着一些问题：1.图像数据是二维的，当图像的分辨率为224^2或者137^2时，

网络的参数数量可能过大 2.图像点与其周围像素之间是有关联的，全连接神经网络不能很好的学习这种关联 3.当网络层数变深时，全连接神经网络的训练会变得很难，从而限制了模型的表达能力。而卷积神经网络则通过参数共享、下采样等等的方式解决上述问题。如图 2-5 展示了一种卷积神经网络的结构。它由卷积层、池化层和全连接层组成。

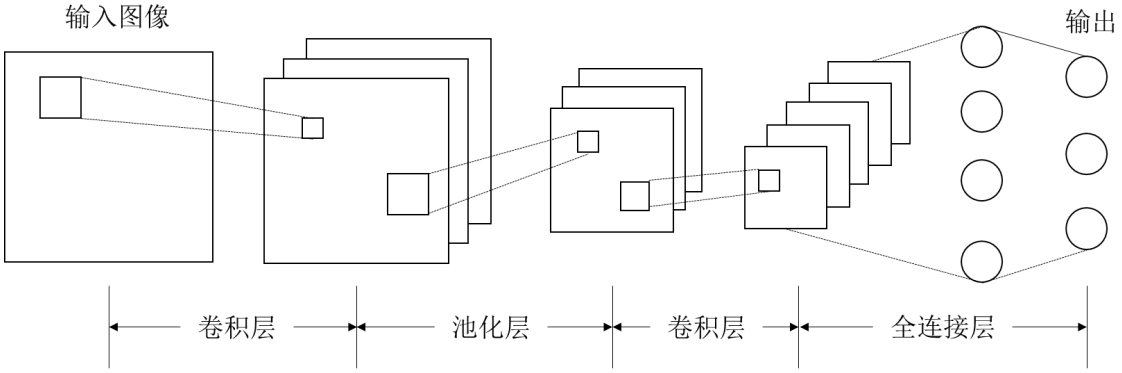

图 2-5　一种卷积神经网络

卷积层的输入是三维的，分别代表着长、宽、通道数，单个卷积核的计算过程可以表示为：

$$y(m,n)=\sum_{k=0}^{K-1}\sum_{i=0}^{I-1}\sum_{j=0}^{J-1}x^{k}(m+i,n+j)H^{k}(i,j)$$

其中 K 代表输入层的通道维度，x^k 是输入层第 k 个通道的数值，H^k 代表卷积核的第 k 个通道的数值，I, J 为输入的长和宽，y 代表经过单个卷积核作用后的输出。如果对于输出的需求是多通道的，则只需要利用多个卷积核计算结果并将结果在通道维度上堆叠起来。这样，就完成了卷积层的计算。

上述卷积层的计算过程是经过简化的，事实上，卷积层可以通过调整一些超参数来使结果变得不同：不同卷积核的大小可以学习到图像不同尺度上的特征；步长的大小也决定了卷积层感受野的大小；是否填充输入的边界决定了卷积层对边界信息的利用程度。这些都可能成为影响结果的因素，在实际构建模型的过程当中应当合理的去选择。

从计算过程中可以看到，卷积层中的单个特征输出不再与每一个的输入所相关，同时，卷积层可以共享卷积核的权重，这两个因素都大大的减少了卷积层的计算量。而且，卷积层的计算考虑到了图像每个像素与其周围像素之间的关系，利用了图像中的位置信息，这也有助于提取图像的特征。

池化层的主要作用是下采样。池化层可以分为平均值池化、最大值池化，它可以将卷积层提取出来的主要特征保留，并将一些冗余的信息去除，如图 2-6 展示了步长为 2, 卷积核大小 2×2 的最大值池化和平均值池化过程。池化层不

改变输入的通道数。

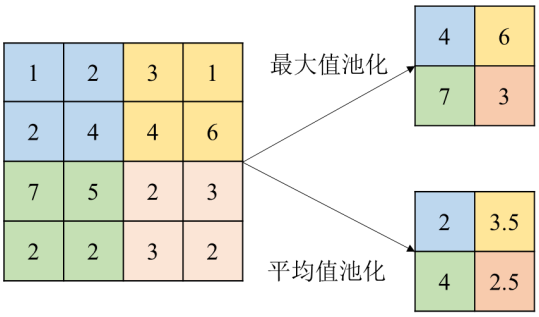

图 2-6　最大值池化和平均值池化

卷积神经网络中可以利用 Relu 函数[32]作为激活函数：

$$\mathrm{Relu}(x)=\max(0,x)$$

该激活函数的计算简单，能够缓解梯度消失问题。

通过卷积层、池化层的计算，卷积神经网络很大程度上的缓解了全连接神经网络处理图像问题时参数数量过大的问题，同时，随着网络层的变深，它可以学习到图像不同层次的特征，浅层的卷积层更多的学习图像的纹理特征，深层的卷积层则更多的去学习图像的几何细节特征，通过利用这些特征，卷积神经网络可以高效的完成许多计算机视觉中的任务。

2.2.3　注意力机制

在注意力模型出现之前，研究人员在利用深度学习完成机器翻译、语音识别等等任务时，常常出现输出与输入之间无法对齐的情况：当想要利用一个句子逐个翻译其中单词时，每个输出单词都是利用同样的句子经过编码、解码这一过程获得的，而显然输入与输出对应单词之间的关系比较大，应当更多的利用对应单词去完成翻译任务。注意力机制就是考虑到了上述的情况，该机制模拟人脑的运行机制，通过学习来关注输入中的重点部分，而对无关部分选择性的忽略。而注意力机制经过几年来的发展，不仅仅可以应用到自然语言处理领域当中，而且在计算机视觉领域也取得了很大的成果。如图 2-7 展示了一种注意力模块的计算过程。

具体的，给定输入 X 对应的键(key)：

$$K=\{k_1,k_2,\cdots,k_N\}$$

N 代表输入序列的长度，以及对应的值(value):

$$V=\{v_1,v_2,\cdots,v_N\}$$

首先计算每次查询(query) q 与键之间的注意力得分：

$$w_i = s(k_i, q),\ i = 1,2,\cdots,N$$

其中 s 为计算相似性的函数，常用的有点乘公式：

$$s(k_i, q) = k_i^{\mathrm{T}} q$$

添加权重公式：

$$s(k_i, q) = k_i^{\mathrm{T}} W q$$

先拼接之后添加权重的公式：

$$s(k_i, q) = W[k_i; q]$$

其中 W 代表对应的参数矩阵。得到注意力得分之后，利用 Softmax 函数对其进行归一化操作得到权重：

$$\alpha_i = \mathrm{Softmax}(w_i) = \frac{e^{w_i}}{\sum_{j=1}^{N} e^{w_j}}$$

最后，根据权重矩阵得到对应的注意力输出：

$$\mathrm{Att}(q, X) = \sum_{i=1}^{N} \alpha_i v_i$$

这样，就完成了注意力机制的一次计算过程。

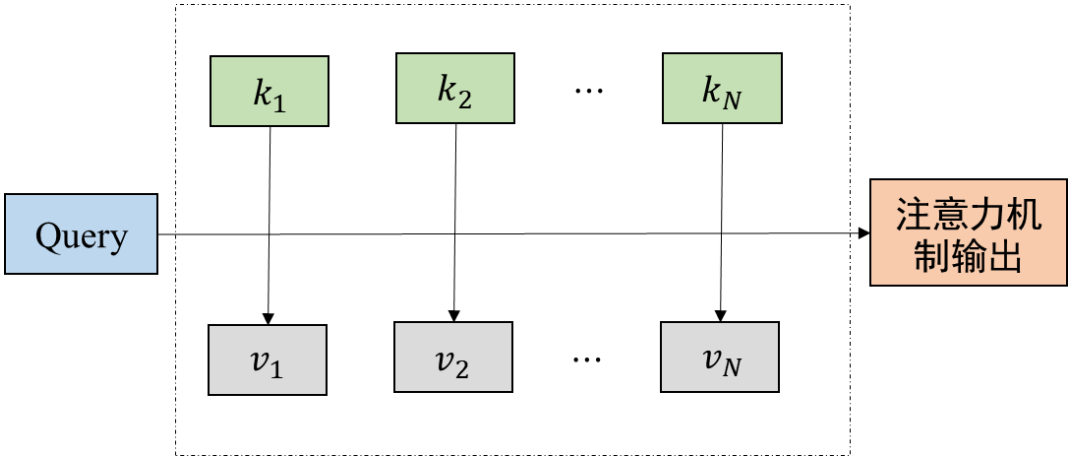

图 2-7　一种注意力模块的计算过程

注意力机制也有很多的变种，比如说硬性注意力，它直接选取最高概率的信息，而不进行权重的融合。但是该种方法无法求对应的梯度值，因此在神经网络中也就无法进行训练。

另一种常用的变种是多头注意力机制，具体地，设查询 Q 为：$Q = \{q_1, q_2, \cdots, q_M\}$，其中 M 代表查询的数量。多头注意力机制的计算公式为：

$$\mathrm{Att}(Q, X) = \mathrm{Att}(q_1, X) \oplus \mathrm{Att}(q_2, X) \oplus \cdots \oplus \mathrm{Att}(q_M, X)$$

其中 $\oplus$ 代表深度学习中的拼接操作，它利用 M 个不同的查询平行的计算多次注

意力模块，提取输入中的不同部分信息。

2.3 卷积神经网络模型的发展

经过近些年来的发展，许多卷积神经网络模型诸如 GoogLeNet[33], ResNet[34], Yolo[35]系列, U-net[36]等受到了广泛的关注。这些经典的网络从各方面的优化了处理图像的过程，并且对本文网络模型的设计有很强的启发作用，因此本节将对其中两种经典的网络进行简要介绍。

2.3.1 残差神经网络

在残差神经网络出现之前，人们在训练神经网络的时候经常遇到的一个问题是越来越深的网络并不能使结果变得更好。因为在人们普遍的认知当中，越深的神经网络参数越多，相应的，模型的泛化能力也应该越强，模型的表现也应当更好，但当人们将卷积神经网络的层数加到 20 层之后，网络收敛较慢、结果也变差了。ResNet 可以很好的解决这一问题，它将残差的思想应用到了设计模型当中。如图 2-8 展示了残差模块。

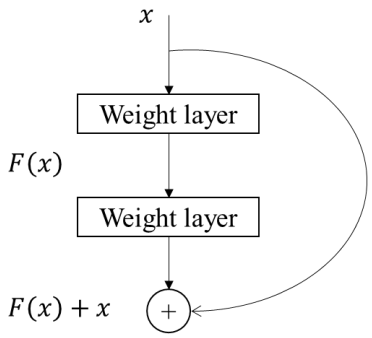

图 2-8 残差模块示意图[34]

残差模块的核心其实就是恒等映射，加入恒等映射之后输出可以表示为：

$$y = F(x) + x$$

其中 $F(x)$ 是残差部分。而如果输出与输入之间的通道数目不一致，则可以通过卷积操作来调整通道数，表示为：

$$y = F(x) + W(x)$$

其中 W 代表卷积操作。同时，ResNet 网络对于深层结构，特别的提出了一种瓶颈(bottleneck)结构来降低计算量，如图 2-9 所示。

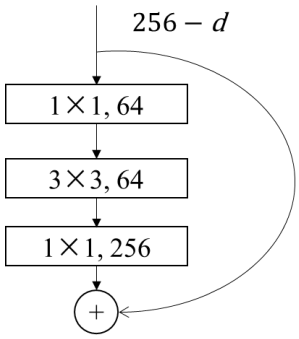

图 2-9　bottleneck 结构示意图[34]

该瓶颈结构先将输入通过大小为1×1的卷积核降低通道数，再通过3×3的卷积核提取特征，最后再利用1×1的卷积核恢复通道数。利用上述的残差结构和瓶颈结构，研究人员在设计模型的过程中就可以考虑将卷积层堆叠的更深，从而能在许多图像领域的任务中获得更好的效果。

2.3.2　U-net 模型

U-net 模型在图像语义分割的任务中取得了良好的表现。如图 2-10 完整的显示了 U-net 的模型结构。

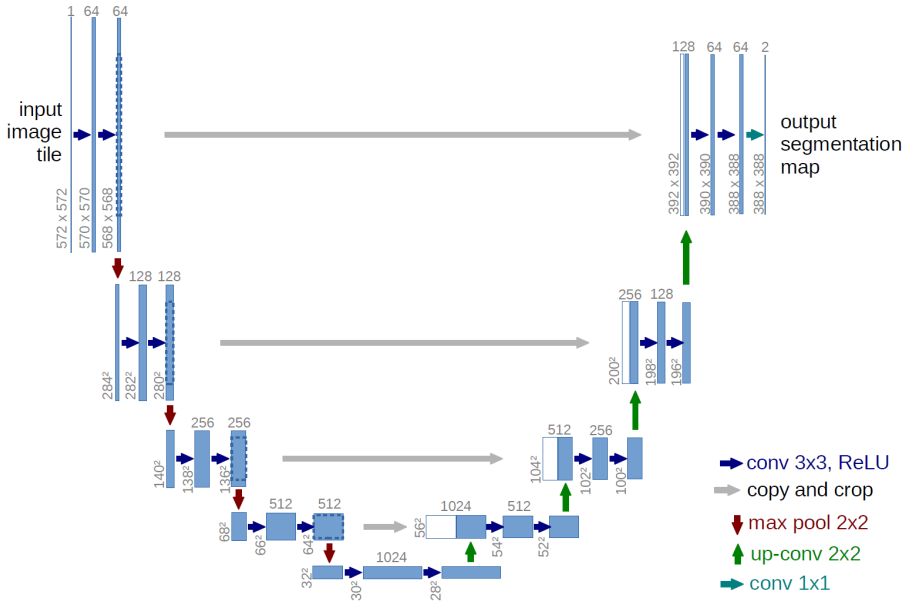

图 2-10　U-net 网络结构[36]

U-net 首先利用卷积层和池化层对输入图像做连续四次的下采样操作，再进行四次上采样。并且其利用跳越连接的方式将对应层的输出在通道维度拼接起来，这样做的好处在于同时利用了浅层次的分辨率信息以及深层次的语义信息，前者能够提供图像中精细特征，后者则可以提供目标的整体特征。跳跃连

接的过程中可能存在两个信息维度不一致的问题，因此需要将原信息裁剪成对应的形状。其下采样采用的是最大值池化操作，保留了主要特征。由于其结构为 U 型，因此称为 U-net 模型。

对于 ResNet 中的残差连接和 U-net 中的跳跃连接来说，其主要的区别在于前者是直接意义上的将两个层次中的信息相加，而后者则是将两个层次中的信息拼接起来。这两种形式互有优劣，前者的表示形式更加效率，直接融合两层之间的特征；后者则更细致的保留了两层的具体特征信息。总的来说，区别于不同的图像任务当中，两者的表现是不一样的，残差连接适合于图像识别这种需要提取深层次特征的任务，跳跃连接适用于语义分割这种需要保持图像分辨率的任务。

2.4 本章小结

本章首先介绍了相机将三维世界中的物体映射到二维像素空间的过程，接着对全连接神经网络、卷积神经网络和注意力机制的计算过程进行了讲解，最后简要介绍了两种经典的卷积神经网络模型：ResNet, U-net. 并对两个模型的关键结构进行了比较，分析了它们之间的类似之处以及区别，根据它们的特点说明了它们适用的应用场景。

第 3 章　三维重建物体体素

3.1　引言

三维数据在计算机中有很多种的表达方式。因此，研究人员在完成三维重建任务时可以选择不同的表达形式。现阶段，体素是应用较为广泛的一种形式。体素类似于图像的形式，主要区别在于：1.体素是用来描述三维空间物体的，而图像则是二维的。2.不同于图像像素点的值代表的是颜色信息，体素中值为 1 代表该点是占据状态，值为 0 代表该点为空状态。体素的优点是表示比较直观，易于计算机处理。

三维重建又分为单视图重建和多视图重建。单视图重建就是利用单张图像重建出物体的三维结构，而多视图重建则可以利用物体不同角度的图像完成重建。对于单视图重建这一任务来说，设计模型需要从以下的几个方面去考虑：1.如何效率的提取图像特征。2.如何利用图像特征推测出物体不可见部分的三维结构。3.如何平衡好体素分辨率与模型复杂度之间的关系。因为单一图像势必会缺失很多物体的信息，所以想要完成单视图重建的任务，就必须提取出图像的纹理、结构等等特征信息，并使用这些信息完成可视部分与不可视部分的重建。体素的分辨率大小也是影响结果的重要因素，分辨率过小，则重建的物体会比较模糊，分辨率过大，模型的复杂度会很高，因此，也需要选用合适的分辨率来平衡重建质量与模型复杂度。

多视图重建则需要考虑的是体素融合的问题。相对于单视图重建来讲，多视图重建可以利用到的信息更多，从不同角度图像获得到的特征是不一样的，它们分别描述了物体不同角度的信息，如何将这些特征整合利用是设计模型的关键。同时，多视图重建任务的数据量也会变大，图像之间存在着一些相似的结构信息，因此，模型也应当具备去除冗余信息，提取各个图像中关键信息的能力。这不仅仅会提升重构的效果，同时也会减少模型的计算复杂度。

基于以上几点的考虑，本章将基于神经网络建立模型，在单视图和多视图的情况下完成体素重建。

3.2　单视图三维重建模型

本文提出了一种基于视觉 transformer 的三维重建模型，如图 3-1 展示了网络的整体结构。

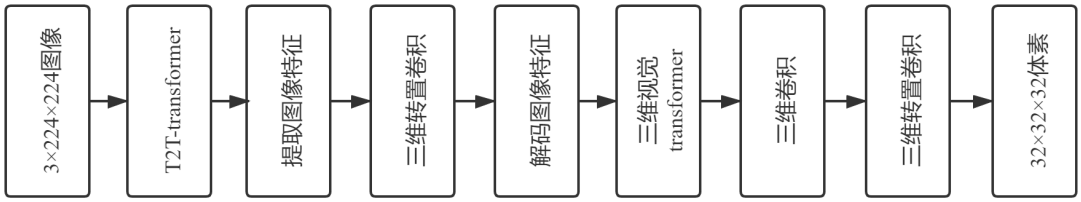

图 3-1　本文三维重建体素模型整体结构

总体上来说，本文模型使用单个 RGB 图像作为网络的输入，通过基于视觉 transformer 的编码层提取图像不同维度之上的特征，再通过三维反卷积将提取到的特征解码并得到粗略的体素信息，最后通过三维视觉 transformer 模块和三维卷积模块提升体素的精度，最后输出分辨率为 32^3 的体素。接下来本文将详细的介绍其中各个模块。

3.2.1　网络的编码层

网络的编码层应当具有提取图像特征的能力，过去的几年中，卷积神经网络在这一领域应用广泛，并且效果优异。而随着 transformer 模块[37]在自然语言处理领域的成功，一些研究人员尝试将这些方法应用到计算机视觉[38,39]当中去，并且也取得了不错的效果。Li 等人就基于这种想法提出了 T2T-transformer 结构，实验结果表明该模型在图像识别的任务上可以取得优于残差神经网络的效果，因此，本文基于该结构设计三维重建模型的编码层。

如图 3-2 展示了 T2T-transformer 模块的具体结构。

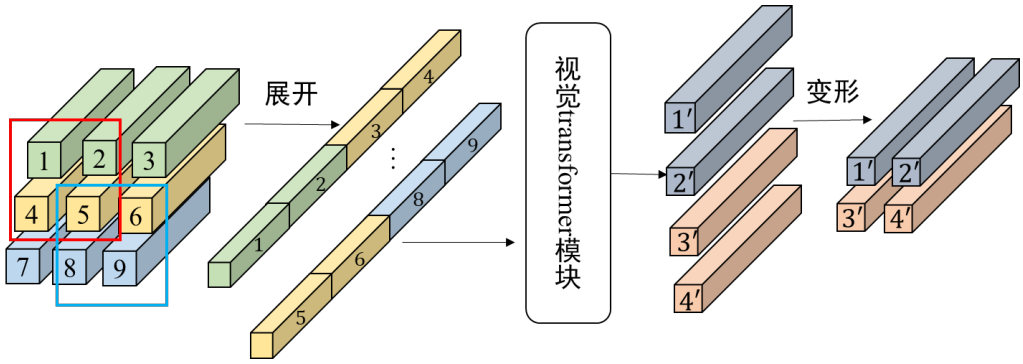

图 3-2　T2T-transformer 结构

该结构首先将图像转换成图像块的形式：假设该层的输入为 $x \in R^{C\times H\times W}$，其中 H,W 代表输入的分辨率，C 代表输入的特征维度。接着，按照特定的滑动窗口顺序提取图像块，假设滑动窗口的大小为 $l\times l$，则每一个图像块 x_p 的大小就为 $l\times l\times c$，假定滑动的步长为 s，则图像块的总数目为 $\left((H-l)/s+1\right)\times\left((W-l)/s+1\right)$，在这里简记为 N．这样，再将每个图像块扁平化

为 $x_p^{'} \in R^{D\times 1}$，经过上述步骤，输入的尺度就变为了：

$$x' \in R^{N\times D}$$

$x^{'}$ 可以看作是图像块的序列，接下来对每个图像块引入自注意力机制。该机制模拟的是人们观察物体时的视觉注意力机制，通常人们认知物体时并不会从头到尾的仔细观察它，而是根据不同的需求和场景观察其中特定的部分，而随着不断的学习，就会察觉到关键部分经常会出现在哪些位置，并将注意力放在上面，从而效率的提取到重要信息。

具体到图像块之上，首先利用全连接层得到图像块自注意力机制中的键(key) x_k，查询(query) x_q，值(value) x_v:

$$x_k = x'W_k, W_k \in R^{D\times D'}, x_k \in R^{N\times D'}$$

$$x_q = x'W_q, W_q \in R^{D\times D'}, x_q \in R^{N\times D'}$$

$$x_v = x'W_v, W_v \in R^{D\times D'}, x_v \in R^{N\times D'}$$

其中 $D^{'}$ 代表新的特征维度。接着，利用矩阵点乘计算 query 和 key 之间的相似性得到权重矩阵：

$$x_w = x_q \cdot x_k^{\mathrm{T}}$$

接着使用 Softmax 函数归一化权重矩阵：

$$x_w = \mathrm{Softmax}\left(x_w\right) = \frac{\exp\left(x_w\right)}{\sum_{i=0}^{N}\sum_{j=0}^{N}\exp\left(x_{w_{ij}}\right)}$$

最后将权重矩阵和 value 值相乘得到结果：

$$x_{\mathrm{att}} = x_w \cdot x_v$$

这样，每一个图像块都可以通过注意力机制的模块提取到对应的 $D^{'}$ 维的特征。因此经过注意力层的计算，$x^{'}$ 的维度就从 $N\times D$ 变成了 $N\times D^{'}$.

接下来使用神经网络的多层感知力机制，该层由多层全连接层以及 dropout 层来组成，利用 Gelu 函数作为激活函数，定义为：

$$\mathrm{Gelu}(x) = xP(x \le X) = x\Phi(x)$$

X 是具有零均值和单位方差的高斯随机变量。这里的 $\Phi(x)$ 是正态分布的概率分布函数。Gelu 函数引入了随机正则的思想，并在一定程度上可以减少梯度消失的问题。如图 3-3 显示 Gelu 函数的图像。

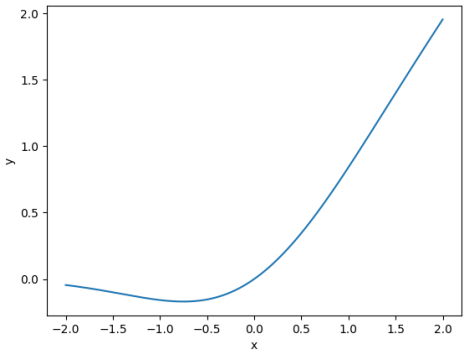

图 3-3　Gelu 函数图像，横轴代表输入值，纵轴代表输出值

通过引入高斯误差函数：

$$\mathrm{erf}\left(x\right)=\frac{2}{\sqrt{\pi}}\int_{0}^{x}e^{-t^{2}}\mathrm{d}t$$

其中 x 代表输入值。Gelu 函数就可以通过计算高斯误差函数来计算：

$$\mathrm{Gelu}\left(x\right)=\frac{1}{2}x\left(1+\mathrm{erf}\left(\frac{x}{\sqrt{2}}\right)\right)$$

在实际计算的过程中可以通过近似高斯误差函数来求得 Gelu 函数的近似值。Gelu 函数的导数可以表示为：

$$\frac{\mathrm{d}}{\mathrm{d}x}\mathrm{Gelu}\left(x\right)=\phi\left(x\right)+x\phi'\left(x\right)=\phi\left(x\right)+xP\left(x=X\right)$$

这样，通过神经网络的多层感知力机制，就得到了每个图像块的新特征 $x_o\in R^{N\times D'}$，接下来将图像块特征重塑到空间维度：

$$I=\mathrm{Reshape}\left(x_o\right)$$

这里具体的操作是将特征矩阵 $x_o\in R^{N\times D'}$ 转换成图像分辨率的模式 $I\in R^{D'\times H'\times W'}$，其中 H',W',D' 分别代表输出的长、宽、特征维度。且有 $N=H'\times W'$.

通过上述的几个过程，编码层就学习到了图像之中的特征。可以看到，视觉 transformer 结构不仅仅考虑到了图像中每个像素点与其周围像素点的联系，同时也通过对每个图像块引入自注意力机制学习图像块中的重点特征。具体的，本文编码层利用四个视觉 transformer 模块提取特征，滑动窗口的大小分别为 7×7，3×3，3×3，3×3，接着利用一个 2×2 的最大值池化保留主要特征，最终学习到 $256\times8\times8$ 大小的特征信息。

3.2.2　网络的解码层

解码层的目的是将编码层学习到的图像特征信息转换成粗略的体素信息。

而图像经过编码层学习之后特征维度变大，但是尺度信息变小，因此需要将尺度因子进行放大。这里就需要用到上采样的方法，上采样常用的方法有三种：反池化、双线性插值、转置卷积。转置卷积能一定程度上恢复分辨率信息和特征信息，并且有可学习的参数，因此本文模型使用三维转置卷积来解码特征，并生成粗略的体素信息，便于之后多视图重建任务使用。

普通的单层卷积操作将多值映射到单值，假设输入的通道数为 2, 卷积核的大小为3×3, 则每次卷积都将输入中的 18 个值映射到单值，这个过程一般来讲会减小分辨率。而转置卷积则是希望将输入的分辨率变大。转置卷积类似于在输入周围填充元素，将其在分辨率维度扩展，再进行卷积的过程。

具体的，本文模型首先将图像编码层学习到的大小为$256\times8\times8$的特征重塑成$2048\times2\times2\times2$维度，接着采用五个转置卷积层进行解码，输入的特征维度分别降低到 512, 256, 128, 32, 1, 卷积核的大小为$4\times4\times4$, 最后一层利用 Sigmoid 激活函数输出体素的概率值。

3.2.3 网络的输出层

通过解码层可以得到粗略的体素信息，但是结果比较模糊，需要进一步地重构结果。本文在二维视觉 transformer 结构的基础之上，提出一种三维的视觉 transformer 结构用于重构体素。

具体的，设输入$x\in R^{C\times H\times W\times L}$，其中$H,W,L$分别代表输入的长、宽、高维度，$C$代表特征维度。接着，利用三维的滑动窗口提取体素模块，并将其在特征维度堆叠，设滑动窗口的大小为$l\times l\times l$，则每个提取到的体素块尺寸为$x_p\in R^{D\times1}$, $D=l\times l\times l\times c$, 再将每个图像块按照顺序排列形成矩阵得到三维体素块的特征信息矩阵：

$$x'\in R^{N\times D}$$

其中$N=\left(\left(H-l\right)/s+1\right)\times\left(\left(W-l\right)/s+1\right)\times\left(\left(L-l\right)/s+1\right)$, 代表三维体素块的数目。这样，就得到了类似于二维视觉 transformer 的特征矩阵。区别是三维体素块代表了三个方向上的特征。类似的，计算三维体素块中的键、查询、值，并最终获得特征输出。同时，在模型的输出层中也引入了残差模块提升精度、防止过拟合。具体的做法是将转置卷积前后同一尺度的特征信息直接相加作为下一层的输入，这样的做法保留了一些原始特征信息。

本文在三维卷积层后引入了 Elu 函数作为激活函数去提升重建效果，该激活函数具有单侧饱和性的特点。

Relu 作为激活函数有许多的优点：反向传播中梯度的值只能为 0 或 1，可以一定程度上的缓解梯度消失的问题；具有单侧饱和性，在朝着输入值减少的方向，输出值会趋于 0，这会减少噪声信息的干扰；具有稀疏性，梯度计算较快。

但是，由于 Relu 函数的稀疏性，其也会带来神经元死亡的问题：如果神经元的取值为负的话，其输出值就为 0，在后续的传播过程中就不会起到作用，并且其反向传播的过程中梯度也为 0，对应的参数也就会无法更新，这样，该神经元就始终会在负侧不起作用。LeakRelu[40]函数的提出一定程度上解决了上述问题。LeakRelu 函数的定义为：

$$\text{LeakRelu}(x)=\begin{cases} x & x>0 \\ \alpha x & x\le 0,\alpha=0.1 \end{cases}$$

α 为超参数，x 代表输入值。该函数的导数可以写为：

$$\frac{\mathrm{d}}{\mathrm{d}x}\text{LeakRelu}(x)=\begin{cases} 1 & x>0 \\ \alpha & x\le 0 \end{cases}$$

其就是在 Relu 激活函数的一侧引入了一个“泄露值”，LeakRelu 激活函数的图像如图 3-4 所示。

有较多种方式可以对超参数 α 进行选择。一种常用的方法是取常数值，可以取 0.1 或者是 0.01. 也可以对参数 α 取随机值，其分布满足标准的正态分布函数，该函数称为随机 LeakRelu 函数。也可以将 α 看作可以学习的参数，在神经网络的训练过程当中对其进行更新，该函数被称为 PRelu[41]激活函数。

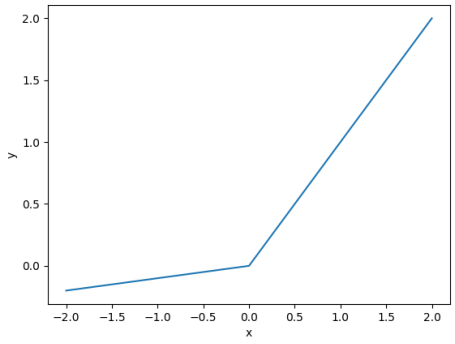

图 3-4　LeakRelu 函数图像，横轴代表输入，纵轴代表输出

但是，LeakRelu 激活函数也存在着一定的局限，它放弃了 Relu 激活函数的单侧饱和性质。Elu 函数则可以克服上述的缺点，其定义为：

$$\text{Elu}(x)=\begin{cases} x & x>0 \\ \alpha\left(e^{x}-1\right) & x\le 0 \end{cases}$$

超参数 α 的取值一般为 1. 其梯度计算公式为：

$$\frac{\mathrm{d}}{\mathrm{d}x}\mathrm{Elu}(x)=\begin{cases}1 & x>0\\ \alpha e^{x} & x\le 0\end{cases}$$

如图 3-5 展示了超参数 α 取值为 1 情况下 Elu 激活函数的图像。

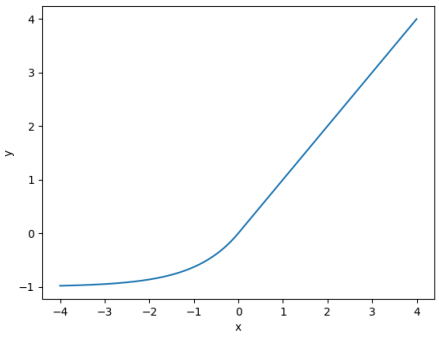

图 3-5　Elu 激活函数图像，横轴代表输入，纵轴代表输出

Elu 激活函数在朝着输入值减少的方向，输出值趋于 $-\alpha$. 因此该激活函数具有单侧饱和的性质。

具体的，本文使用一个滑动窗口大小为 $4\times4\times4$ 的三维视觉 transformer 模块和两个卷积核大小为 $4\times4\times4$ 的三维卷积模块重构体素结果，并利用三维转置卷积恢复体素的分辨率，最终输出大小为 $32\times32\times32$ 的体素概率值。

3.2.4　网络的正则化层

为了提高网络的稳定性以及加快网络的收敛速度，本文的模型引入了批正则化[42]和层正则化的策略。批正则化是将每次训练中的批次数据在特征维度上做归一化。首先来看全连接层上的批正则化，设训练时的 batch size 为 N，该层输入数据的特征维度为 D，首先求得输入在特征维度上的均值和方差：

$$E_k=\frac{1}{N}\sum_{i=1}^{N}x_{i,\mathrm{k}}$$

$$\sigma_k^2=\frac{1}{N}\sum_{i=1}^{N}\left(x_{i,k}-E_k\right)^2$$

其中，$x_{i,k}$ 代表第 i 个数据的第 k 个特征，E_k 和 σ_k^2 分别代表第 k 个特征的均值以及方差。接下来对每个值做归一化操作：

$$\hat{x}_{i,k}=\frac{x_{i,k}-E_k}{\sqrt{\sigma_k^2+\varepsilon}}$$

其中 ε 为常数防止方差为 0. 这种直接的表达方式可能会丢失一部分原先的特

征信息，因此通过引入可学习的参数 β,γ 来解决这一问题：

$$y_{i,k}=\gamma_k\hat{x}_{i,k}+\beta_k$$

γ_k 和 β_k 分别代表第 k 个特征的可学习参数。

批正则化也可以应用于二维卷积层以及三维卷积层。设二维卷积层的输入尺寸为 $B\times C\times H\times W$，其中 B 代表 batch size，C 代表特征维度，H，W 代表输入的分辨率，则二维卷积的正则化层就是在特征维度之上作归一化，每个特征层求得一个对应的均值、方差以及可学习的参数 γ_k，β_k. 类似的，三维卷积层也可以使用对应的批正则化策略。

层正则化的策略则与批次的大小无关，批正则化主要是对不同数据中同一类型的特征进行正则化操作，层正则化是对同一数据中的不同特征做归一化，这是两者的主要区别。

设全连接层输入的特征数目为 D，首先分别求出尺寸维度上的均值以及方差：

$$E=\frac{1}{D}\sum_{i=1}^{D}x_i$$

$$\sigma^2=\frac{1}{N}\sum_{i=1}^{N}\left(x_i-E\right)^2$$

x_i 代表输入数据的第 i 个特征。接着对数据做尺度维度上的归一化操作：

$$\hat{x}_i=\frac{x_i-E}{\sqrt{\sigma^2+\varepsilon}}$$

同样的，引入可学习参数 β,γ 保留特征信息：

$$y_i=\beta\hat{x}_i+\gamma$$

这样，就完成了层正则化的操作。

本文模型在卷积层之后使用批正则化的操作，在视觉 transformer 中利用层正则化的操作来加速模型的训练以及提高模型的稳定性。

3.3 多视图三维重建模型

多视图三维重建体素任务的重点在于如何效率的利用不同角度的图像去重建物体。因此，设计模型需要从以下的几点去考虑：1.对于多个角度图像的特征提取应当是高效的 2.当输入图像的顺序不同时，重建的结果差距不应该较大。3.尽可能地利用物体在图像中的可视部分重建结果。多视图中各个图像的视点是不同的，对于物体在图像中的可见部分，其三维重建的结果较好，反之，三

维重建的结果较差。因而本文利用图像的特征学习对应的体素权重，并根据权重完成融合。

具体的，设从不同图像学习到的三维特征集合为$S=\{x_1,x_2,\cdots,x_N\}$，其中$x_n\in R^{C\times H\times W\times L}$代表第$n$个图像所学到的特征，接着利用三维卷积层学习注意力得分$C=\{c_1,c_2,\cdots,c_N\}$，其中：

$$c_n=g(x_n),\ c_n\in R^{H\times W\times L}$$

这里g可以代表多个三维卷积操作。接下来将得到的注意力得分通过 Softmax 函数正则化成注意力权重$S=\{s_1,s_2,\cdots,s_N\}$，其中：

$$s_n^{(h,w,l)}=\frac{\exp\left(c_n^{(h,w,l)}\right)}{\sum_{i=1}^{N}\exp\left(c_i^{(h.w.l)}\right)}$$

$s_i^{(h,w,l)}$代表第i个权重在(h,w,l)位置处的值。再设各个图像通过单视图三维重建生成的体素概率为$V=\{v_1,v_2,\cdots,v_N\}$，其中$v_n\in R^{H\times W\times L}$，接着将权重与对应的体素相乘并相加形成融合后的体素y：

$$y=\sum_{i=1}^{N}s_i*v_i,y\in R^{H\times W\times L}$$

其中$*$代表按元素相乘操作。该过程利用特征生成了权重，利用注意力机制的想法赋予图像重建效果好的地方较大的权重，使各个图像产生的体素概率值能够更好的融合。用于提取图像特征的卷积层以及生成注意力权重的三维卷积层参数都是共享的，在图像输入的数目不同时，该方法能自适应的利用这些共享参数进行体素融合。而对于不同序列的图像该方法的重建效果是一致的，因为经过融合后(h,w,l)处体素的概率值可以写为：

$$y^{(h,w,l)}=\sum_{i=1}^{N}v_i^{(h,w,l)}*s_i^{(h,w,l)}=\sum_{i=1}^{N}v_i^{(h,w,l)}*\frac{\exp\left(c_i^{(h,w,l)}\right)}{\sum_{i=1}^{N}\exp\left(c_i^{(h.w.l)}\right)}=\frac{\sum_{i=1}^{N}v_i^{(h,w,l)}*\exp\left(c_i^{(h,w,l)}\right)}{\sum_{i=1}^{N}\exp\left(c_i^{(h.w.l)}\right)}$$

由上式可知，计算体素融合后的值是利用相加求和的方法，与各个图像的输入顺序无关。

具体的，本文采用五个卷积核大小为$3\times3\times3$的三维卷积层从特征中提取权重，在网络的解码层之后添加该基于三维卷积的注意力模块，根据该模块融合各个图像生成的体素概率值，并最终利用网络输出层输出融合后的体素结果。

3.4 实验结果与分析

本节将通过实验验证模型的可行性。在单视图和多视图的情况下说明了本文模型在 ShapeNet 数据集下的效果，并和主流方法进行对比。

3.4.1 实现细节与评估标准

神经网络模型通过最小化损失函数来更新参数，损失函数设计的好坏能够直接影响模型的输出结果。本文模型使用体素情况下的二元交叉熵损失函数，可以公式化的写为：

$$l=-\frac{1}{N}\sum_{i,j,\mathrm{k}}\left(y_{(i,j,k)}\log\left(p_{(i,j,k)}\right)+\left(1-y_{(i,j,k)}\right)\log\left(1-p_{(i,j,k)}\right)\right) \tag{3-1}$$

其中 N 代表体素块的总数量，$y_{(i,j,k)}$ 代表体素块在 (i,j,k) 处的真实值，值为 1 或者 0. $p_{(i,j,k)}$ 代表预测的体素概率值，损失函数越小，代表预测值与真实值之间越接近。

ShapeNet 数据集是依据 WorldNet[43]层次组织的三维 CAD 模型，本文采用 3D-R2N2 中提供的 13 种类型的 43736 个三维模型作为数据集。将其中的 32700 个模型划分为训练集，剩下的划分为测试集。同时，3D-R2N2 也给出了每个模型的 24 张随机视角的渲染图，使用这些图片作为网络的输入。如图 3-6 展示了 ShapeNet 中的渲染图和对应体素的例子。

图 3-6　ShapeNet 中渲染图和体素例子，类型分别为飞机和显示器

本文使用 IoU 作为评价指标，它能描述预测体素与真实体素之间的差距：

$$\text{IoU}=\frac{\sum_{i,j,\text{k}} I(p_{(i,j,k)}-t)I(gt_{(i,j,k)})}{\sum_{i,j,\text{k}} I[I(p_{(i,j,k)}-t)+I(gt_{(i,j,k)})]}$$

$p_{(i,j,k)}$ 和 $gt_{(i,j,k)}$ 分别代表着在 (i,j,k) 处体素的预测值和真实值，t 代表体素阈值。$I(\cdot)$ 代表指示函数，在这里其定义为：

$$I(x)=\begin{cases}1 & x>0\\ 0 & x\le 0\end{cases}$$

更高的 IoU 值代表更好的重建结果。

在训练网络时，本文使用 224×224 分辨率大小的图像作为输入，batch size 设置为 16, 输出的体素分辨率为 $32\times32\times32$，利用 Adam 方法作为优化器，学习率设置为 0.001, 在 7 次迭代之后变为 0.0005. 单视图重建一共训练了 10 次。多视图重建在单视图重建权重的基础上，输入多个图像训练网络，再训练 2 次。利用 pytorch 实现了本文的网络。

3.4.2 单视图重建结果

表 3-1 本文模型与其他模型在测试集下的 IoU 结果对比

类别名	本文结果	3D-R2N2[19]	OGN[44]	PSGN[24]	AttSets[45]
飞机	0.635	0.513	0.587	0.601	0.594
长凳	0.555	0.421	0.481	0.550	0.552
储藏柜	0.783	0.716	0.729	0.771	0.783
汽车	0.849	0.798	0.828	0.831	0.844
椅子	0.551	0.466	0.483	0.544	0.559
显示器	0.534	0.468	0.502	0.552	0.565
灯	0.455	0.381	0.398	0.462	0.445
扬声器	0.710	0.662	0.637	0.737	0.721
步枪	0.608	0.544	0.593	0.604	0.601
沙发	0.724	0.628	0.646	0.708	0.703
桌子	0.604	0.513	0.536	0.606	0.590
手机	0.733	0.661	0.702	0.749	0.743
船	0.610	0.513	0.632	0.611	0.601
平均值	0.652	0.560	0.596	0.640	0.642

表 3-1 展示了本文模型的实验结果，本章中体素阈值 t 取为 0.4. 可以看到，

不同类别的 IoU 值有较大差距。汽车和橱柜类型的重建效果较好，因为其类型中的物体结构比较相近，便于网络去推测其结构，而灯和显示器这两种类型的重建效果一般，一方面是由于物体的不可见部分较多，另一方面也是因为物体结构的差异比较多，不利于模型推测其结构。

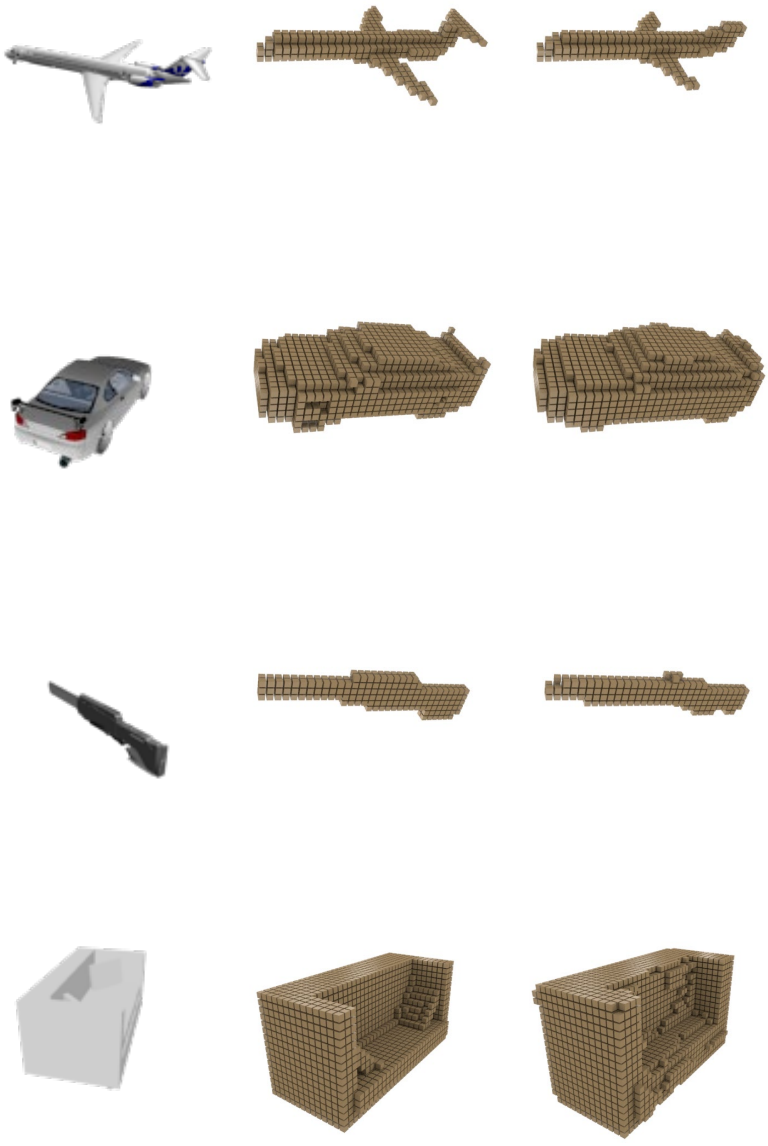

图 3-7　单视图三维重建体素例子。从左到右依次为输入图像，目标体素，本文模型重构体素，从上到下的类型分别为飞机、汽车、步枪、沙发

表 3-1 也展示了本文模型与其他模型对比的实验结果，OGN[44]使用八叉树模型来表示目标体素，它可以在有限的内存空间中表示更高分辨率的 3D 输出，但是，它的表示比较复杂，而且随着分辨率的提高，网络训练会变得困难。可

以看到，本文模型在飞机、汽车这种实例的类型中重建效果很好，本文实验结果对比 PSGN 等模型的结果也有优势。

图 3-7 给出了本文模型三维重建的实例，分别有飞机、汽车、沙发、步枪类型，可以看到，本文模型能够利用单幅 RGB 图像恢复具有物体整体结构的体素模型，也能够对飞机的机翼、汽车的轮胎、沙发的靠垫等等特殊结构进行重建。

3.4.3 多视图重建结果

多视图重建可以利用更多的信息，表 3-2 对比了本文模型在不同输入图像数目下三维重建结果的 IoU 值。可以看到，随着视点的增加，模型在测试集上的效果也不断地变好。说明模型应用了多个图像的特征去重建体素。相对于单视图结果而言，多视图的重建结果有了明显提升。比如说对于椅子这种类型，加入一张图像就可以使得 IoU 值提高 0.3, 效果比较明显。

表 3-2 本文模型在多视图情况下重建体素的 IoU 值

类别名	1-view	2-view	3-view	4-view	5-view
飞机	0.6347	0.6510	0.6592	0.6627	0.6644
长凳	0.5550	0.5752	0.5789	0.5829	0.5845
储藏柜	0.7825	0.7922	0.7949	0.7955	0.7956
汽车	0.8488	0.8576	0.8607	0.8619	0.8631
椅子	0.5506	0.5802	0.5884	0.5910	0.5931
显示器	0.5343	0.5514	0.5704	0.5699	0.5741
灯	0.4552	0.4663	0.4755	0.4774	0.4764
扬声器	0.7098	0.7235	0.7299	0.7313	0.7316
步枪	0.6082	0.6270	0.6288	0.6317	0.6309
沙发	0.7235	0.7413	0.7467	0.7462	0.7452
桌子	0.6043	0.6171	0.6241	0.6259	0.6275
手机	0.7326	0.7668	0.7721	0.7738	0.7783
船	0.6095	0.6297	0.6306	0.6328	0.6325
平均值	0.6516	0.6683	0.6743	0.6762	0.6773

表 3-3 展示了本文多视图重建结果与其他方法的结果对比，结果显示本文模型能够在输入图像数目不同时，较好的重建出物体的体素模型。并且本文模

型对比 3D-R2N2 模型在重建速度上有所提升，利用更少的参数数量重建出了更好的结果。并且本文多视图模型能够自适应的利用任意数量的图像完成重建任务，比 3D-R2N2 模型更加灵活。

表 3-3　多视图重建下本文模型与其他模型的 IoU 结果对比

	1-view	2-view	3-view	4-view
Attsets[45]	0.642	0.662	0.670	0.675
3D-R2N2[19]	0.560	0.603	0.617	0.625
本文模型	0.652	0.668	0.674	0.676

图 3-8 给出了本文模型三视图情况下重建的具体实例。可以看到，在多视图的情况之下，能够对灯这种细长的物体也有较好的重构结果，也能重构出桌腿这种易被遮挡的结构。

图 3-8　三视图重建例子，前三幅图像为模型输入图像，接下来依次为目标体素，模型重构体素。从上到下物体的类型分别为灯、桌子、手机、飞机

3.5 本章小结

本章首先介绍了物体体素结构的特点，同时说明了单视图和多视图体素重建问题中的关键点。之后分别从编码层、解码层、输出层、正则化层详细的讲解了模型是如何在单视图的情况下获得体素的。接着利用注意力机制的想法将各个不同图像的三维重建结果融合起来，完成多视图体素重建。最后，给出了本文模型的实验结果，表明了本文模型能够恢复出具有物体结构的体素，并且较 3D-R2N2、OGN 等方法重建效果有提高。

第 4 章　三维重建物体网格

4.1　引言

上一章生成的体素模型可以看出物体的大致形状以及主要结构，但是它仍然存在着一些缺点。首先是分辨率的问题，它生成的体素模型的大小是 $32\times32\times32$ 的，对于细节的描述不足；其次它也没有考虑顶点与顶点之间的关联性。多边形网格这一表示方法则可以一定程度上的克服上述缺点。

多边形网格是由不规则的顶点、边以及面组成。如果其中每个面是三角形，则被称为三角网格，是多边形网格中较常见的一种形式，可以抽象的表示成集合的形式 $M=\{V,E,F\}$，其中 $V=\{v_i\}_{i=1}^{N}$，v_i 代表着第 i 个点的顶点坐标，$E=\{e_i\}_{i=1}^{M}$，e_i 代表着第 i 条边包含的两个顶点对应的索引，$F=\{f_i\}_{i=1}^{P}$，f_i 代表着第 i 个面所包含的三个顶点对应的索引。三角网格的分辨率大小基本不受限制，同时其边和面能够反映出三维物体的线条特征，能够更加细节的帮助人们去理解物体的三维结构。

由于其数据的不规则形，深度学习中传统的卷积网络无法直接应用到网格数据当中去。图卷积网络则可以解决这一问题，它能对图数据进行学习以及更新，将三角网格中的每个顶点看作是图卷积中的节点，边上对应的顶点看作是邻接节点，就可以将三角网格这一数据结构应用到图卷积神经网络当中去。

而由二维图片生成三维网格这一过程是十分困难的，因此许多方法中使用一个初始化的网格结构，比如说正方体网格、椭球体网格，利用这些初始化的结构变形成理想的结构。但是这类方法在物体结构与初始化结构差距较大的情况下重建效果不理想。因此，本章考虑首先利用卷积层学习图像特征并生成粗略的体素模型，将体素模型按照一定的规则转换成粗糙的三维网格模型，并基于图卷积神经网络对网格进行变形，输出较精确的三维网格模型。

4.2　网格重构模型

如图 4-1 展示了重构网格的整体模型，它的输入是单张的图像，首先通过 ResNet-50 网络学习图像特征，接着通过三维卷积模块提取三维特征，并生成粗略的体素模型，接着将体素模型转换为三维网格模型。在变形三维网格的过程中，将卷积层学习到的二维特征以及三维特征投影到网格顶点上，并利用图

卷积神经网络变形三角网格，最终输出结果。接下来本节将对其中的每个模块进行详细的介绍。

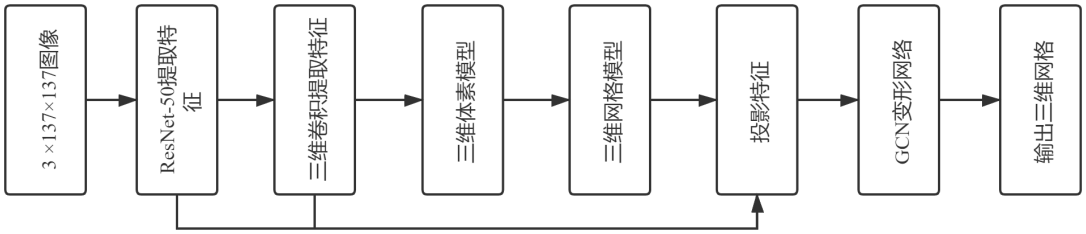

图 4-1　本文网格重构模型总体结构

4.2.1　初始化网格模型

不同于许多方法中使用特定形状的网格作为模型的输入，本文利用输入图像生成粗略的体素模型，并根据该体素模型生成初始化的网格，该网格能够粗略的体现物体的结构。

本章中生成粗略体素的方法类似于第三章，在效率上进行了一定的优化。首先，大小为137×137 的图像通过 ResNet-50 网络依次提取四个层次上的特征，特征大小分别为：256×56×56，512×28×28，1024×14×14，2048×5×5．将这些二维特征保存下来后续使用。接着，将最后一个特征利用第三章的解码层进行解码，并利用三维卷积在其中提取三维特征，提取到的三维特征经过拼接后的大小为：36×32×32×32，并且利用这些三维特征生成体素模型。上述过程不仅仅承担着重建体素的任务，而且其提取到的图像特征会投影到网格各个顶点之上，便于图卷积网络利用顶点特征更新顶点的坐标。

接下来利用立方化方法由体素生成网格。首先将体素概率大于阈值 t 的位置都变成由三角网格组成的立方体，每个立方体由 8 个顶点、18 条边、12 个面组成。为了形成网格表面，还需要消除存在于网格内部的面。具体的操作如表 4-1 所示。该算法满足了生成网格速度上的要求，另一种常用的算法是移动立方体算法(Marching Cubes)，该方法通过计算等值面的方法生成网格，可视化结果较好，但是由于其计算量比较大因此形成网格的速度比较慢，不适用于本文网络。

这样，就生成了一个初始化的网格。该网格顶点的数目与体素分辨率的大小息息相关，本文生成的粗略体素分辨率大小为32×32×32，生成的顶点数目范围为[600, 1400]. 后续网络通过移动这些顶点来变形网格，在这个过程当中，维持了顶点之间边和面的关系。

表 4-1　由体素生成表面网格过程

输入：体素概率值 $V:[0,1]^{H\times W\times L}$，体素概率阈值 t

for $(h,w,l)\in \text{Range}(H,W,L)$:
　IF $V[h,w,l]>t$ then:
　　在 (h,w,l) 处添加三角网格立方体
　　IF $V[h-1,w,l]>t$ then:
　　　移除后面位置的两个面
　　END IF
　　IF $V[h+1,w,l]>t$ then:
　　　移除前面位置的两个面
　　END IF
　　IF $V[h,w-1,l]>t$ then:
　　　移除左边位置的两个面
　　END IF
　　IF $V[h,w+1,l]>t$ then:
　　　移除右边位置的两个面
　　END IF
　　IF $V[h,w,l-1]>t$ then:
　　　移除上边位置的两个面
　　END IF
　　IF $V[h,w,l+1]>t$ then:
　　　移除下边位置的两个面
　　END IF
　END IF
融合共享顶点
输出：三维网格模型 $M=\{V,F,E\}$

4.2.2　投影特征

在利用图卷积网络变形网格之前，还有重要的一步需要考虑，就是顶点的特征怎样去得到。在生成体素的过程中模型提取到了不同层次的二维特征以及三维特征，本文模型利用双线性插值以及三线性插值的方法将这些特征与三角网格中的顶点联系起来。

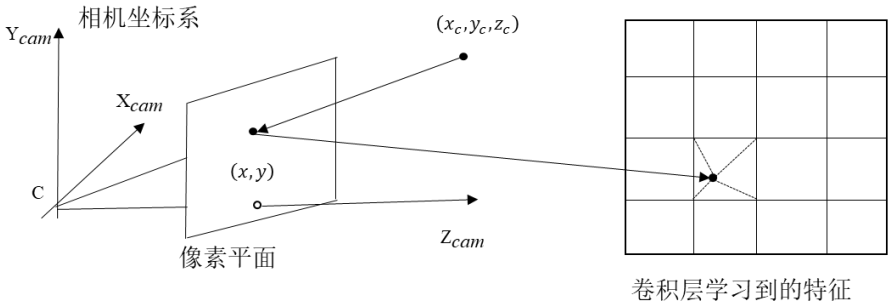

图 4-2　投影图像特征到对应的网格顶点

如图 4-2 展示了二维特征投影的过程。三维网格的顶点坐标是相对于相机坐标系而言的，首先要做的是将相机坐标系的点$\{v_c\}$转换到像素坐标系上：

$$v = Kv_c$$

其中，K代表相机的内参矩阵。接着，利用双线性插值的方法得到该点的二维特征：

$$f = \frac{(x_2-x)(y_2-y)}{(x_2-x_1)(y_2-y_1)}f_{1,1} + \frac{(x-x_1)(y_2-y)}{(x_2-x_1)(y_2-y_1)}f_{2,1} + \frac{(x_2-x)(y-y_1)}{(x_2-x_1)(y_2-y_1)}f_{1,2} + \frac{(x-x_1)(y-y_1)}{(x_2-x_1)(y_2-y_1)}f_{2,2}$$

其中$f_{1,1}$代表$v_i^{'}$左上角位置的特征，不同的$f_{i,j}$定义类似。(x,y)代表顶点坐标，每个点对应的二维特征数取决于之前图像编码层学习到的特征维度。

同样的，也可以利用三线性插值的方法获得顶点的三维特征。不同的是，三维特征和网格顶点都是在相机坐标系上的，直接插值即可：

$$f = \sum_{i,j,k=1}^{2} \frac{|x_{3-i}-x||y_{3-j}-y||z_{3-k}-z|f_{i,j,k}}{(x_2-x_1)(y_2-y_1)(z_2-z_1)}$$

其中，$f_{i,j,k}$代表对应位置的特征。对于超过像素边界或者网格边界的点。其特征被设为 0.

再把每个顶点上的二维特征以及三维特征拼接起来，就得到了每个顶点最后的特征。这样，每个顶点的特征包含不同尺度下的二维特征以及三维特征，变形网格模块可以利用到的信息更多。

4.2.3　变形网格

在得到每个三角网格中顶点的特征之后，本节利用带有残差连接的图卷积神经网络(GCN)变形网格，输出最后的网格结果。

GCN 是作用于图数据上的卷积。考虑一个简单无向图$G=(V,E)$，其中有N个顶点M条边。设v_i代表图上第i个节点，D代表图上的度矩阵，A代表图上的邻接矩阵，则该无向图的拉普拉斯矩阵就可以写为：

$$L = D - A$$

对称归一化之后的拉普拉斯矩阵就可以写为：

$$L^s = D^{-\frac{1}{2}} L D^{-\frac{1}{2}} = I - D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$$

它是半正定的对称矩阵，因此可以谱分解为：

$$L^s = U \Lambda U^{-1}$$

Λ 是由其特征值组成的对角矩阵，U 为正交矩阵，因此上式可以改写为：

$$L^s = U \Lambda U^T$$

则 U^T 就可以看成是傅里叶变换基，U 是逆傅里叶变换基，那么图上的傅里叶变换就可以写成：

$$F\{f\}=U^T f$$

对应的，图上的傅里叶逆变换就可以写为：

$$F^{-1}\{f\}=Uf$$

而由卷积定理 f 和 x 的卷积可以写为：

$$f * x = F^{-1}\{F\{f\} \cdot F\{x\}\}$$

这里的 $*$ 代表卷积操作。那么，在图上的卷积就可以写为：

$$(f * x)_G = U\left(\left(U^T f\right) \odot \left(U^T x\right)\right)$$

其中 $\odot$ 代表哈达玛积。f 在应用当中可以代表滤波函数，类似于二维卷积，f 应该影响一个节点周围的相邻节点，因此可以把 f 定义为关于拉普拉斯矩阵的函数 $f(L)$，每作用一次拉普拉斯矩阵就相当于在图上传播一次邻居节点，进一步的，将 $U^T f$ 化为对角矩阵 $f_\theta(\Lambda)$，参数是 θ，这样，就可以将上式改写：

$$f_\theta * x = U f_\theta U^T x \tag{4-1}$$

式(4-1)的计算复杂度要求很高，主要原因在于需要求拉普拉斯矩阵的特征向量，利用多项式展开对 f_θ 近似：

$$f_\theta(\Lambda) \approx \sum_{k=0}^{K} \theta_k T_k(\Lambda)$$

T_k 这里代表切比雪夫多项式。设 $K=1$，卷积公式可以简化为：

$$f_\theta * x \approx \theta_0 x + \theta_1 L x$$

假设 $\theta_0 = 2\theta$，$\theta_1 = -\theta$，则有：

$$f_\theta * x \approx \theta(I_N + D^{-\frac{1}{2}} A D^{-\frac{1}{2}})x$$

反复使用该模型可能会造成神经网络梯度不稳定的问题，因此引入一个正则化

策略。令 $\tilde{A}=A+I_N$，$\tilde{D}_{ii}=\sum_j \tilde{A}_{ij}$，则上述迭代公式被替代为：

$$f_\theta * x \approx \theta(\tilde{\mathrm{D}}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}})x$$

加上激活函数σ，就可以得到计算简化后的图卷积公式：

$$H^{(l+1)}=\sigma(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)})$$

上述介绍的图卷积网络能够很好的利用节点之间的关联去学习特征，但是，GCN 无法直接应用深层结构。有关的实验表明，GCN 在二层的时候效果最好，再往上堆叠层数反而使得模型的效果变差。这一问题被称为过度平滑。而浅层的 GCN 无法提取节点的高阶特征。使用类似于残差连接的策略也仅仅只是能缓解过度平滑的问题，不能使得深层结构比二层的结构表现得更好。基于上述的考虑，许多的学者提出了一些新的结构[46,47]使图卷积网络能变得更深。

Chen[48]等人提出了一种带有初始化残差连接和恒等映射的图卷积网络，其迭代公式可以写为：

$$H^{(l+1)}=\sigma\left(\left(\left(1-\alpha_l\right)\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}+\alpha_l H^{(0)}\right)\left(\left(1-\beta_l\right)I_n+\beta_l W^{(l)}\right)\right)$$

$H^{(0)}$代表图中节点的初始特征信息，α_l和β_l代表两个超参数。残差连接确保了每个节点的最后表示保留了输入层的部分信息。该公式不但利用了节点的初始特征，而且将权重矩阵与单位矩阵相加。在网络层数加深的情况之下，该模型效果变好，优于了传统的浅层图卷积模型。

具体的，本文使用网格情况下的图卷积层[26]结合残差连接和恒等映射的策略学习特征，图卷积层的个数为 12, 每经过四层利用特征更新顶点位置，并根据新的位置重新投影图像特征。最终输出网格中的顶点坐标以及网格之中的边对应的顶点索引。

4.2.4 网络的损失函数

在训练体素的过程中，损失函数沿用上一章的体素交叉熵损失函数。公式如(3-1)所示。

在训练网格的阶段本文利用目标网格与预测网格之间的点的差距来定义损失函数。首先，从目标网格和预测网格的表面采样 5000 个点，设P代表目标点云集合，Q代表预测点云集合。计算它们之间的最邻近顶点：

$$\Lambda_{P,Q}=\left\{\left(p,\arg\min_q \|p-q\| : p\in P\right)\right\}$$

P和Q之间的 Chamfer 损失就可以写为：

$$L_{\text{cham}}(P,Q)=|P|^{-1}\sum_{(p,q)\in\Lambda_{P,Q}}\|p-q\|^2+|Q|^{-1}\sum_{(q,p)\in\Lambda_{Q,P}}\|q-p\|^2$$

其中$|P|$，$|Q|$分别代表P，Q中点的个数，Chamfer 损失约束了两个点云之间顶点的距离，但是它没有考虑到网格之中的边，因此需要定义边的损失函数：

$$L_{\text{edge}}(V,E)=\frac{1}{|E|}\sum_{(v,v')\in E}\|v-v'\|^2$$

$E\subseteq V\times V$代表预测网格之中的边的集合，最终本文模型的损失函数将由上述的三个损失函数加权组成。

4.3 实验结果与分析

本章仍然使用 ShapeNet 数据集，不同的是最后生成的是三角形网格模型，同时，在训练和测试的过程中均使用了不同图像对应的相机内参以及外参矩阵。接下来本章将对模型的实验结果进行详细的介绍。

4.3.1 评估标准与实现细节

本文采用 F-Score[49]评估标准，给出距离阈值d，公式化的可以写为：

$$\text{F-Score}(d)=\frac{2P(d)R(d)}{P(d)+R(d)}$$

其中$P(d)$和$R(d)$分别代表着关于距离阈值d的精确率和召回率。可以被计算为：

$$P(d)=\frac{1}{n_R}\sum_{r\in R}\left[\min_{g\in G}\|g-r\|<d\right]$$

$$R(d)=\frac{1}{n_G}\sum_{g\in G}\left[\min_{r\in R}\|g-r\|<d\right]$$

n_R和n_G分别代表预测点云和目标点云的数目。对于 F-Score 这一评估标准来说，值越大代表重建效果越好。

本章使用分辨率为137×137的单张 RGB 图像作为训练以及测试的输入，提取图像特征的网络为在 ImageNet[50]上预训练过的 ResNet-50, 接下来利用三维卷积和三维转置卷积生成分辨率大小为$32\times32\times32$的体素，并将其转成三角网格模型。在变形模块共使用 12 个图卷积网络层变形网格。利用 Adam 优化器训练 15 轮，学习率设置为 0.0001. batch size 大小设置为 4, 体素阈值设为 0.15. 损

失函数的权重设置为：$\lambda_{\text{voxel}}=1,\lambda_{\text{cham}}=1,\lambda_{\text{edge}}=0.2$．本章模型利用 pytorch 以及 pytorch3d 库实现。

4.3.2 重构网格结果

表 4-2 展示了不同物体类型下本模型重建网格的 F-Score 值。距离阈值 $d=0.0001$.

表 4-2 不同阈值 d 下本文模型在测试集上的 F-Score 值

类别名	F-d	F-$2d$
长凳	67.23	78.16
椅子	65.15	78.26
灯	60.40	71.89
扬声器	60.06	73.76
步枪	65.01	73.59
桌子	75.28	84.57
船	54.18	66.84
飞机	74.07	83.73
储藏柜	72.15	84.25
汽车	67.52	81.71
显示器	60.75	74.56
沙发	61.43	76.86
手机	76.93	87.47
平均值	67.65	79.55

可以看到，本文模型对枪、飞机类型重建效果较好，而对灯、扬声器这种类型的重建效果一般。可能的原因是枪、飞机类型的网格比较规则，遮挡表面较少，模型容易得出顶点之间的关联性。而灯这种类型的物体形状比较细长，并且物体与物体之间的差距较大，不利于模型完成重建，因此结果相对来说较差。

同时，表 4-3 也将本文结果也与其他近年来的其他的重建方法进行了对比，以便于说明本文模型的效果。其中，N3MR[51]是一种基于弱监督生成网格模型的方法。由表 4-3 可知，本文模型取得了较好的重建结果。在不同的距离阈值 d 的情况下本文模型较表中几种算法重建结果均有提升。

表 4-3　不同阈值 d 下本文模型与其他模型 F-Score 值的对比

	F-d	F-2d
N3MR[51]	33.80	47.72
3D-R2N2[19]	39.01	54.62
PSGN[24]	48.58	69.78
Pixel2Mesh[26]	59.72	74.19
本文模型	67.65	79.55

如图 4-3 展示了本模型重构网格的几个实例，可以看到，模型重构网格能得到与目标网格类似的结果，并且能够描述纹理、线条等等细节特征。

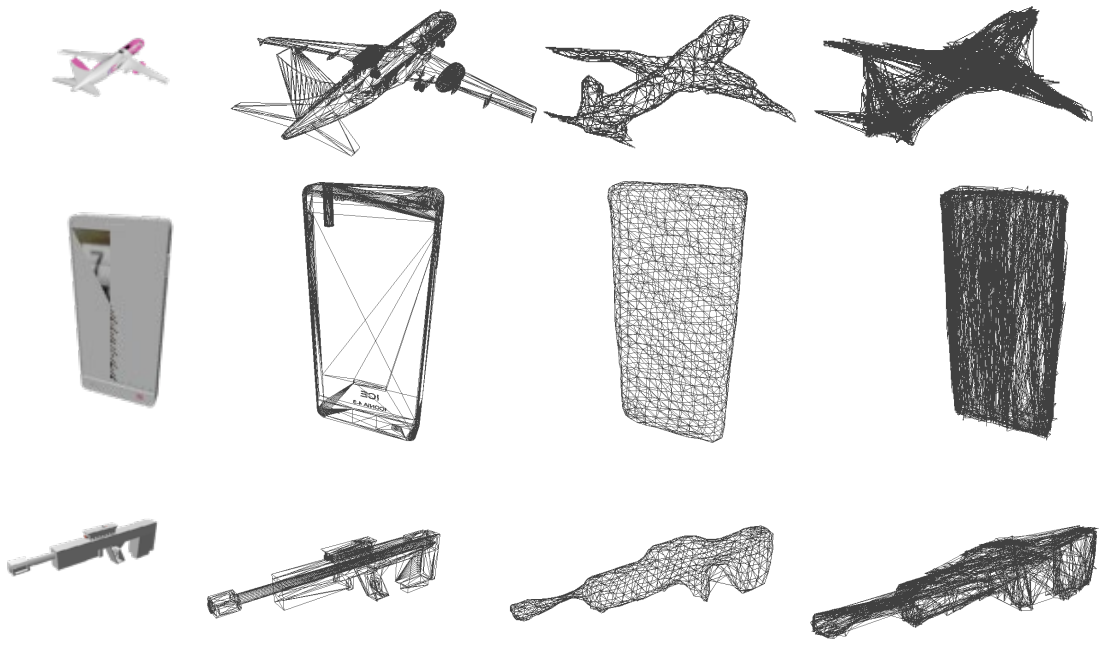

图 4-3　重构网格实例，从左到右依次为输入图像，目标网格，预测网格，去掉 L_{edge} 损失函数后模型的预测网格

有关研究表明去掉 L_{edge} 损失函数能够使得评估标准表现的更好，本文也对此进行了实验。如表 4-4 展示了将 L_{edge} 的权重设为 0, 其余设置不变的结果。图 4-3 也展示了去掉 L_{edge} 损失函数后的对比实验结果。

可以看到，去掉 L_{edge} 损失函数的情况下评价指标确实会变的更好，但是，该损失约束了边的长度，如果去掉该损失函数，出现了许多不规则以及重复的面，这些网格就会显示的不清晰。因此，本文模型依然添加了 L_{edge} 损失函数去

训练模型，以便生成感官效果更好的三角网格。

表 4-4　去掉 L_{edge} 损失后，不同阈值 d 下本文模型在测试集上的 F-Score 值

类别名	F-d	F-2d
长凳	74.26	85.17
椅子	67.93	80.86
灯	64.19	75.42
扬声器	63.64	77.93
步枪	80.16	88.04
桌子	79.56	88.58
船	69.52	81.15
飞机	77.46	86.67
储藏柜	74.52	86.99
汽车	75.93	88.71
显示器	67.15	80.28
沙发	67.10	82.44
手机	78.21	88.82
平均值	73.48	84.94

4.4　本章小结

本章首先分析了三角网格表示方法的特点。接着在单视图的情况下，利用卷积网络生成体素模型，并将该体素模型利用立方化的方法转换成网格模型，接着利用带有残差连接和恒等映射的图卷积网络对网格进行变形，输出结果。最后，对本文模型的结果进行了分析，并和其他方法相比较，说明了本文模型的效果。同时，讨论了不使用 L_{edge} 损失函数下本文模型的表现。

结　论

人们认知客观物体的一种重要手段就是利用视觉推断它的三维模型。对于计算机视觉也是如此，普通的二维图像缺失了许多物体的重要信息，因此在一些情况下需要利用三维重建技术还原出物体的三维结构。三维重建技术在医学医疗、虚拟现实、自动驾驶等等领域都有着重要的应用。而从三维物体映射到二维图像的过程损失了许多信息，因此想要通过图像还原出物体的结构需要利用图像的先验知识去推断未知部分，而神经网络具有强大的表达能力和推断能力。因此，本文利用神经网络构建了三维重建模型，主要内容有：

1.本文设计了一种具有编码器-解码器结构的神经网络模型完成了单视图重建体素的任务，该模型首先通过基于视觉 transformer 的图像编码层提取图像的结构信息，接着利用基于三维转置卷积的解码层恢复出体素分辨率并且保留关键特征，最后利用三维视觉 transformer 模块去输出体素概率值。最后通过在测试集上的表现说明了模型的效果，模型可以在信息缺失的情况下推断物体的未知结构，通过模型生成的体素可以表现出原物体的整体结构。

2.对于多视图三维重建体素任务，本文利用基于三维卷积的注意力模块去融合不同图像重建体素的概率值，该模型能够并行的处理任意数目的图像输入，并且重构的结果与图像输入的顺序无关。在测试集上与单视图重建体素结果相对比，说明了多视图重建模型利用了不同角度图像的信息效率的完成了重建，最后将本文结果与其他模型结果相对比说明了本文模型的效果。

3.本文根据体素这一表达方式的不足，进一步的利用图卷积网络重构出物体的三角网格模型。本文模型首先通过基于 ResNet-50 和三维卷积的编码层获取输入的二维特征以及三维特征，并将重构出的体素模型通过立方化的方法效率的转换成三角网格，将编码层学习到的特征通过线性插值的方式投影到对应顶点之上，最后利用带有残差连接和恒等映射的图卷积网络变形网格，并通过 Chamfer 损失和一种边损失函数训练网络，在约束顶点距离的同时考虑到了网格中边的关系。通过模型在测试集上的数据表明，本文模型重建的三角网格能够表现出物体的线条、纹理等等细节特征，可视化的效果相比于体素更加清晰。

总的来说，本文模型重建体素的速度较快，而本文重建的网格精度较高，能够反映出物体更多的细节和线条，它们可以适用于不同的应用场景。

参考文献

[1] Zhang M L, Desrosiers C. High-Quality Image Restoration Using Low-Rank Patch Regularization and Global Structure Sparsity[J]. IEEE Transactions on Image Processing, 2019, 28(2): 868-879.

[2] Zhang J, Zhao D B, Gao W. Group-Based Sparse Representation for Image Restoration[J]. IEEE Transactions on Image Processing, 2014, 23(8): 3336-3351.

[3] Wang B, Gao X, Tao D C, et al. A Nonlinear Adaptive Level Set for Image Segmentation[J]. IEEE Transactions on Cybernetics, 2014, 44(3): 418-428.

[4] Best-Rowden L, Jain A K. Longitudinal Study of Automatic Face Recognition[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2018, 40(1): 148-162.

[5] Baka N, Kaptein B L, De Bruijne M, et al. 2D-3D Shape Reconstruction of The Distal Femur From Stereo X-ray Imaging Using Statistical Shape Models[J]. Medical Image Analysis, 2011, 15(6): 840-850.

[6] Thomas B H. Virtual Reality for Information Visualization Might Just Work This Time[J]. Frontiers in Robotics and AI, 2019, 6: 84.

[7] Silberman N, Hoiem D, Kohli P, et al. Indoor Segmentation and Support Inference from RGBD Images[C]. European Conference on Computer Vision, 2012: 746-760.

[8] Defferrard M, Bresson X, Vandergheynst P. Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering[C]. Conference on Neural Information Processing Systems, 2016, 29.

[9] Kipf T, Welling M. Semi-supervised Classification with Graph Convolutional Networks[C]. International Conference on Learning Representations, 2017.

[10] Criminsi A. Shape from Texture: Homogeneity Revisited[C]. British Machine Vision Conference, 2000: 82-91.

[11] Zhang R, Tsai P S, Cryer J E, et al. Shape From Shading: A Survey[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1999, 21(8): 690-706.

[12] Ozyesil O, Voroninski V, Basri R, et al. A Survey of Structure from Motion[J]. Acta Numerica, 2017, 26: 305-364.

[13] Lowe D G. Distinctive Image Features from Scale-Invariant Keypoints[J]. International Journal of Computer Vision, 2004, 60(2): 91-110.

[14] Bay H, Tuytelaars T, Van G L. SURF: Speeded Up Robust Features[C].

European Conference on Computer Vision, 2006, 3951: 404-417.

[15] Seitz S M, Curless B, Diebel J, et al. A Comparison and Evaluation of Multi-View Stereo Reconstruction Algorithms[C]. IEEE Conference on Computer Vision and Pattern Recognition, 2006, 519-528.

[16] Newcombe R A, Izadi S, Hilliges O, et al. KinectFusion: Real-Time Dense Surface Mapping and Tracking[C]. IEEE International Symposium on Mixed and Augmented Reality, 2012: 127-136.

[17] Whelan T, Leutenegger S, Salas-Moreno R E, et al. ElasticFusion: Dense SLAM Without A Pose Graph[C]. Robotics: Science and Systems, 2015.

[18] Dai A, Niessner M, Zollhofer M, et al. BundleFusion: Real-Time Globally Consistent 3D Reconstruction Using On-the-fly Surface Reintegration[J]. Acm Transactions on Graphics, 2017, 36(3): 24.

[19] Choy C B, Xu D F, Gwak J Y, et al. 3D-R2N2: A Unified Approach for Single and Multi-view 3D Object Reconstruction[C]. European Conference on Computer Vision, 2016, 9912: 628-644.

[20] Vinyals O, Bengio S, Kudlur M. Order Matters: Sequence to Sequence for Sets[C]. International Conference on Learning Representations, 2016.

[21] Wang M, Wang L J, Fang Y. 3DensiNet: A Robust Neural Network Architecture towards 3D Volumetric Object Prediction from 2D Image[C]. ACM International Conference on Multimedia, 2017: 961-969.

[22] Paschalidou D, Ulusoy A O, Schmitt C, et al. RayNet: Learning Volumetric 3D Reconstruction with Ray Potentials[C]. IEEE Conference on Computer Vision and Pattern Recognition, 2018: 3897-3906.

[23] Xie H Z, Yao H X, Sun X S, et al. Pix2Vox: Content-aware 3D Reconstrction from Single and Multi-view Images[C]. IEEE International Conference on Computer Vision, 2019: 2690-2698.

[24] Fan H Q, Su H, Guibas L. A Point Set Generation Network for 3D Object Reconstruction from A Single Image[C]. IEEE Conference on Computer Vision and Pattern Recognition, 2017: 2463-2471.

[25] Wei Y, Liu S H, Zhao W, et al. Conditional Single-view Shape Generation for Multi-view Stereo Reconstruction[C]. IEEE Conference on Computer Vision and Pattern Recognition, 2019: 9643-9652.

[26] Wang N Y, Zhang Y D, Li Z W, et al. Pixel2Mesh: Generating 3D Mesh Models from Single RGB Images[C]. European Conference on Computer Vision, 2018, 11215: 55-71.

[27] Gkioxari G, Malik J, Johnson J. Mesh R-CNN[C]. IEEE International Conference on Computer Vision, 2019: 9784-9794.

[28] He K M, Gkioxari G, Dollar P, et al. Mask R-CNN[C]. IEEE International Conference on Computer Vision, 2017: 2961-2969.

[29] Mescheder L, Oechsle M, Niemeyer M, et al. Occupancy Networks: Learning 3D Reconstruction in Function Space[C]. IEEE Conference on Computer Vision and Pattern Recognition, 2019: 4455-4465.

[30] Sutskever I, Vinyals O, Le Q V. Sequence to Sequence Learning with Neural Networks[C]. Conference on Neural Information Processing Systems, 2014, 27.

[31] Artetxe M, Labaka G, Agirre E. A Robust Self-learning Method for Fully Unsupervised Cross-lingual Mappings of Word Embeddings[C]. Annual Meeting of the Association-for-Computational-Linguistics, 2018: 789-798.

[32] Nair V, Hinton G E. Rectified Linear Units Improve Restricted Boltzmann Machines[C]. International Conference on Machine Learning, 2010.

[33] Szegedy C, Liu W, Jia Y, et al. Going Deeper With Convolutions[C]. IEEE Conference on Computer Vision and Pattern Recognition, 2015: 1-9.

[34] He K M, Zhang X Y, Ren S P, et al. Deep Residual Learning for Image Recognition[C]. IEEE Conference On Computer Vision and Pattern Recognition, 2016: 770-778.

[35] Redmon J, Divvala S, Girshick R, et al. You Only Look Once: Unified, Real-Time Object Detection[C]. IEEE Conference on Computer Vision and Pattern Recognition, 2016: 779-788.

[36] Ronneberger O, Fischer P, Brox T. U-Net: Convolutional Networks for Biomedical Image Segmentation[C]. International Conference on Medical Image Computing and Computer-Assisted Intervention, 2015, 9351: 234-241.

[37] Vaswani A, Shazeer N, Parmar N, et al. Attention Is All You Need[C]. Annual Conference on Neural Information Processing Systems, 2017, 30.

[38] Ramachandran P, Parmar N, Vaswani A, et al. Stand-Alone Self-Attention in Vision Models[C]. Conference on Neural Information Processing Systems, 2019, 32.

[39] Cao Y, XU J R, LIN S, et al. GCNet: Non-local Networks Meet Squeeze-Excitation Networks and Beyond[C]. IEEE International Conference on Computer Vision, 2019: 1971-1980.

[40] Maas A L, Hannun A Y, Ng A Y. Rectifier Nonlinearities Improve Neural Network Acoustic Models[C]. International Conference on Machine Learning, 2013, 30(1): 3.

[41] He K M, Zhang X Y, Ren S Q. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification[C]. IEEE International

Conference on Computer Vision, 2015: 1026-1034.

[42] Ioffe S, Szegedy C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift[C]. International Conference on Machine Learning, 2015: 448-456.

[43] Miller G A. WordNet: A Lexical Database for English[J]. Communications of the ACM, 1995, 38(11): 39-41

[44] Tatarchenko M, Dosovitskiy A, Brox T. Octree Generating Networks: Efficient Convolutional Architectures for High-resolution 3D Outputs[C]. IEEE International Conference on Computer Vision, 2017: 2017-2115.

[45] Yang B, Wang S, Markham A, et al. Robust Attentional Aggregation of Deep Feature Sets for Multi-View 3D Reconstruction[J]. International Journal of Computer Vision, 2020, 128(1): 53-73.

[46] Wu F, Souza A, Zhang T, et al. Simplifying Graph Convolutional Networks[C]. International Conference on Machine Learning, 2019: 6861-6871.

[47] Xu K, Li C, Tian Y, et al. Representation Learning on Graphs with Jumping Knowledge Networks[C]. International Conference on Machine Learning, 2018: 5453-5462.

[48] Chen M, Wei Z, Huang Z, et al. Simple and Deep Graph Convolutional Networks[C]. International Conference on Machine Learning, 2020: 1725-1735.

[49] Knapitsch A, Park J, Zhou Q Y, et al. Tanks and Temples: Benchmarking Large-Scale Scene Reconstruction[J]. ACM Transactions on Graphics, 2017, 36(4): 78.

[50] Deng J, Dong W, Socher R, et al. Imagenet: A Large-Scale Hierarchical Image Database[C]. IEEE Conference on Computer Vision and Pattern Recognition, 2009: 248-255.

[51] Kato H, Ushiku Y, Harada T. Neural 3D Mesh Renderer[C]. IEEE Conference on Computer Vision and Pattern Recognition, 2018: 3907-3916.

致　谢

行文至此，也就代表着要与培养我六年的哈工大说一声再见了。回首这六年的时间，哈工大带给了我许许多多的感动和快乐。在这里，我向在哈工大遇到的老师、同学、朋友们深深地道一声感谢。

首先感谢我的导师石振锋老师。很荣幸能成为您的学生，在我刚进入课题组的时候，是您无微不至的关心给我前进的勇气与动力。在我对研究方向上迷茫的时候，老师您给我指明了道路，通过您的指导，我对课题有了更为深入的理解。同时，石老师开放和乐观的心态也深深影响了我，每次跟您谈话都感觉轻松而且收获颇丰，教导我们在遇到困难的时候勇敢的去解决，老师也十分关心我们的未来发展，您常常通过您丰富的阅历给予我们人生方向上的指导。您一直是我路上的榜样。

感谢哈工大数学学院的老师们，老师们精彩纷呈的课程给我留下了深刻的印象，我也在您们身上学到了数学的严谨和认真做事的态度，未来的路上我会铭记这份精神，砥砺前行。

感谢我在哈工大遇到的朋友们。课题组的同学们包容互助，我们在一块讨论问题，互相交流想法，解决遇到的困惑与问题，也感谢石老师给我们创造的这个温馨的课题组环境。感谢我的研究生舍友们，我们总是一起庆祝开心的事，我们共同进步，你们的陪伴使我的硕士生活更加丰富，给我留下了很多难忘的回忆。

最后，我要感谢我深爱着的家人们。你们的支持是我不断前进的动力，家永远能够给我温暖和感动，包容我的缺点，在我遇到挫折时鼓励我，在我有所小成时一起分享快乐，只希望未来能有更多的时间陪伴你们。我也会带着你们的期许与鼓励，继续前行。