**计算机图形学课后作业题目**

**学号 162010312 姓名 王康 分数**

1. **试发挥你的想象力，举例说明计算机图形学有哪些应用范围，解决的问题是什么？计算机图形学在构建“元宇宙”中发乎哪些作用？（15分）**

近年来计算机图形学已经广泛地用于多种领域，如科学、医药、商业、工业、政府部门、艺术、娱乐业、广告业、教育和培训等。

其中它的主要应用领域有以下几个：

1、计算机辅助设计与制造(例如：CAD/CAM)

这是计算机图形学最广泛、最重要的应用领域。它使工程设计的方法发生了巨大的改变，利用交互式计算机图形生成技术进行土建工程、机械结构和产品的设计正在迅速取代绘图板加工字尺的传统手工设计方法，担负起繁重的日常出图任务以及总体方案的优化和细节设计工作。

2、计算机辅助教学(例如：CAI)

在这个领域中，图形是一个重要的表达手段，它可以使教学过程形象、直观、生动，激发学生的学习兴趣，极大地提高了教学效果。随着微机的不断普及，计算机辅助教学系统已深入到家庭。

3、计算机动画（例如：flash、Premiere）

通过计算机制作动画，只需生成几幅被称作“关键帧”的画面，然后由计算机对两幅关键帧进行插值生成若干“中间帧”，连续播放时两个关键帧被有机地结合起来。这样可以大大节省时间，提高动画制作的效率。

4、管理和办公自动化

计算机图形学在管理和办公自动化领域中应用最多的是绘制各种图形，所有这些图形均以简明形式呈现出数据的模型和趋势，加快了决策的制定和执行。

5、国土信息和自然资源显示与绘制

国土信息和自然资源系统记录全国的大地和重力测量数据、高山和平原地形、河流和湖泊水系、道路桥梁、城镇乡村、农田林地植被、国界和地区界以及地名等。利用这些存储的信息不仅可以绘制平面地图，而且可以生成三维地形地貌图，为高层次的国土整治预测和决策、综合治理和资源开发研究提供科学依据。

6、科学计算可视化领域

科学计算可视化是利用计算机图形学方法将科学计算的中间或最后结果以及通过测量得到的数据以图形形式直观地表示出来。科学计算可视化广泛应用于气象、地震、天体物理、分子生物学、医学等诸多领域。

7、计算机游戏方面

计算机游戏目前已成为促进计算机图形学研究特别是图形硬件发展的一大动力源泉。计算机图形学为计算机游戏开发提供了技术支持，如三维引擎的创建。建模和渲染这两大图形学主要问题在游戏开发中的地位十分重要。

8、虚拟现实领域

虚拟现实技术的应用非常广泛，可以应用于军事、医学、教育和娱乐等领域，通过带上具有立体感觉的眼睛、头盔或数据手套，通过视觉、听觉、嗅觉、触觉以及形体或手势，整个融进计算机所创造的虚拟氛围中，从而取得身临其境的体验。

9、地理信息系统

地理信息系统是建立在地理图形之上的关于各种资源的综合信息管理系统，是计算机图形学的一个重要应用领域。

10、农业上的应用

借助计算机图形生成技术来保存和再现不同作物种类和不同生长时期的植物形态，模拟植物的生长过程，从而合理地进行选种、播种、田间管理以及收获等。

****计算机图形学在元宇宙中的作用：****  
　　计算机图形学，就是将各种图形通过各种算法，转化为计算机中的数据，在计算机程  
序中形成相应的图形。许多计算机大佬曾经都想象过一种可能：用计算机创造一个虚拟的世界。而想要创建一个虚拟的世界，想要有其真实性，需要还原现实世界中的视觉，听觉，味觉，触觉等等。很显然其中，最容易实现的便是视觉。于是，计算机科学家们便想到了利用计算机来模拟现实的场景。众所周知，任何宏观层面的东西都可以通过点线面来构成。因此，计算机学家们便探索出了各种建模计算。为了使得物体更加真实，计算机学家们又发明出了各种渲染算法。为了让物体动起来，科学家们又通过各种方程，进行图形模拟计算，通过各种运动方程来构建虚拟世界中，物体的运动，使其最大程度接近真实世界。例如，对于刚体，我们有动力学方程，对于柔体，我们又有本构方程；对于流体，我们有Navier—Stokes 方程。至此，我们已经能够通过计算机，形成非常真实的虚拟世界。  
　　元宇宙最大特色，就在于其真实性。它是一个虚拟的宇宙，但在虚拟中有有着各种虚拟的现实物品，元宇宙可以看作现实世界的延申，是现实世界的拓展。由于其真实性，人们在元宇宙中能够做到他们在现实生活中想做的事，甚至是现实中完全不可能的事。而元宇宙中的物品越是拟真，人们在体验元宇宙过程中，所感受到的就越真实，体验就越好，也就越能满足现实中的需求。而计算机图形学正是让元宇宙更加真实的基础。元宇宙中人们看见的所有物品，都是通过计算机图形学的各种算法，各种函数模拟计算出来的。  
　　因此，计算机图形学可以说是元宇宙的基础，是元宇宙的核心技术，对元宇宙起到决定性作用。没有计算机图形学，元宇宙也就无法实现。

1. **OpenGL的主要功能有哪些和OpenGL的绘制流程。（15分）**

**Opengl主要具有七大功能：**

1.建模：OpenGL图形库除了提供基本的点、线、多边形的绘制函数外，还提供了复杂的三维物体（球、锥、多面体、茶壶等）以及复杂曲线和曲面绘制函数。

2.变换：OpenGL图形库的变换包括基本变换和投影变换。基本变换有平移、旋转、缩放、镜像四种变换，投影变换有平行投影（又称正射投影）和透视投 影两种变换。其变换方法有利于减少算法的运行时间，提高三维图形的显示速度。

3.颜色模式设置：OpenGL颜色模式有两种，即RGBA模式和颜色索引（Color Index）。

4.光照和材质设置：OpenGL光有自发光（Emitted Light）、环境光（Ambient Light）、漫反射光（Diffuse Light）和高光（Specular Light）。材质是用光反射率来表示。场景（Scene）中物体最终反映到人眼的颜色是光的红绿蓝分量与材质红绿蓝分量的反射率相乘后形成的颜色。

5:纹理映射（Texture Mapping）。利用OpenGL纹理映射功能可以十分逼真地表达物体表面细节。

6:位图显示和图象增强图象功能除了基本的拷贝和像素读写外，还提供融合（Blending）、抗锯齿（反走样）（Antialiasing）和雾（fog）的特殊图象效果处理。以上三条可使被仿真物更具真实感，增强图形显示的效果。

7:双缓存动画（Double Buffering）双缓存即前台缓存和后台缓存，简言之，后台缓存计算场景、生成画面，前台缓存显示后台缓存已画好的画面。

**Opengl的绘制流程：**

OpenGL绘图的基本过程是：清除缓冲区，设置当前颜色，绘制几何图元，输出图形。

1.清除缓冲区

缓冲区是用来保存绘图过程中产生的图像数据的内存区域，OpenGL有颜色、深度、累积和模板等四种常用的缓冲区。这里只介绍颜色缓冲区和深度缓冲区的清除方法。

由于缓冲区中可能保留有上一次绘图遗留下来的图像数据，这些数据会影响本次绘图，因此在绘制新图之前必须将它们清除掉。颜色缓冲区中存放的是图像的像素颜色数据，深度缓冲区存放的是用于图像消隐的深度信息。在一个三维场景中，距离视点较近的物体可能部分地或全部地遮挡住距离视点较远的物体。将被遮挡的部分小曲不显示的技术成为图像消隐技术。实现图像消隐最简单的方法是使用深度缓冲区(Z-Buffer)，清除深度缓冲区的目的是使消隐工作能正确地进行。一般情况下需要清除的缓冲区就是这两个缓冲区。

清除缓冲区的工作分两步进行。

1)设置清除值。清除值是在清除缓冲区时用来填充缓冲区的数据。设置颜色缓冲区清除值的函数是glClearColor()，它有四个浮点型参数，分别表示清除值的R、G、B、A分量的值，默认值全为0。其中第四个参数A表示Alpha值，主要用于融合，一般把它设为0。如果使用整个窗口作为OpenGL的绘图区域，则可以将颜色缓冲区的清除值看作是窗口的背景颜色。设置深度缓冲区清除值的函数是glClearDepth()，它只有一个参数，即深度清除值，取值范围为[0.01,1.0]，默认值为1.0。

2)清除缓冲区。清除缓冲区使用的函数是glClear()，它只有一个参数，表示将要清除的缓冲区的种类，可取的值有GL\_COLOR\_BUFFER\_BIT(颜色缓冲区)、GL\_DEPTH\_BUFFER\_BIT(深度缓冲区)、GL\_ACCUM\_BUFFER\_BIT(累积缓冲区)、GL\_STENCIL\_BUFFER\_BIT(模板缓冲区)。这些值也可以用逻辑或组合起来使用，实现一次清除多个缓冲区。

2.设置颜色

根据需要设置当前颜色。一旦一种颜色被设置为当前颜色，其后所有物体都将用该颜色绘出，直到当前颜色被设置为新的颜色为止。设置颜色的函数是glColor()。

3.绘制几何图形

OpenGL中用于绘图的函数有很多，这里只介绍基本几何图元的绘图函数。在glBegin()和glEnd()函数之间调用基本图元函数完成点、线、和多边形的绘制。glBegin()函数原型是：void glBegin(GLenum mode);

glEnd()函数原型是：void glEnd(void);

在glBegin()和glEnd()之间，只能调用下表所列的函数，调用其他函数会导致OpenGL错误。glBegin()和glEnd()函数必须成对使用，否则可能导致不可预期的错误。

4.图形输出

glFlush()函数启动OpenGL绘图命令的执行，并将图形光栅化后输出在设备上。如果OpenGL工作在双缓冲模式下，缓冲区切换函数(SwapBuffers())隐含执行glFlush()。

1. **阐述图形与图像有何区别与联系？（10分）**

广义地讲，凡是能在人的视觉系统中形成视觉印象的客观对象均可称为图形。而图像是人类视觉的基础，是自然景物的客观反映，是人类认识世界和人类本身的重要源泉。“图”是物体反射或透射光的分布，“像“是人的视觉系统所接受的图在人脑中所形版的印象或认识，照片、绘画、剪贴画、地图、书法作品、手写汉学、传真、卫星云图、影视画面、X光片、脑电图、心电图等都是图像。图像是客观对象的一种相似性的、生动性的描述或写真，是人类社会活动中最常用的信息载体。或者说图像是客观对象的一种表示，它包含了被描述对象的有关信息。它是人们最主要的信息源。

**两者的区别：**

（1）图形是矢量的概念。它的基本元素是图元，也就是图形指令；而图像是位图的概念，它的基本元素是像素。图像显示更逼真些，而图形则更加抽象，仅有线、点、面 等元素。

（2）图形的显示过程是依照图元的顺序进行的。而图像的显示过程是按照位图中所安排的像素顺序进行的，与图像内容无关。

（3）图形可以进行变换且无失真，而图像变换则会发生失真。例如当图像放大时边界会产生阶梯效应，即通常说的锯齿。

（4）图形能以图元为单位单独进行属性修改、编辑等操作。而图像则不行，因为在图像中并没有关于图像内容的独立单位，只能对像素或图像块进行处理。

（5）图形使用专门软件将描述图形的指令转换成屏幕上的形状和颜色，而图像是将对象以一定的分辨率分辨以后将每个点的信息以数字化方式呈现，可直接快速在屏幕上显示。

（6）图形描述轮廓不很复杂，色彩不是很丰富的对象，如：几何图形、工程图纸、CAD、3D造型软件等。  
而图像表现含有大量细节（如明暗变化、场景复杂、轮廓色彩丰富）的对象，如：照片、绘图等，通过图像软件可进行复杂图像的处理以得到更清晰的图像或产生特殊效果。  
 （7）图形是对图形的控制与再现，而图像是对图像进行编辑、压缩、解压缩、色彩一致性再现等。

**两者的联系：**图形实际上是对图像的抽象。在处理与存储时均按图形的特定格式进行，一旦上了屏幕，它就与图像没有什么两样了。在抽象过程中，会丢失一些原型图像信息。换句话说，图形是更加抽象的图像。

1. **阐述Bresenham画线法的基本原理。（10分）**

Bresenham 画线算法的思想很简单，假设只讨论在第一象限，斜率存在且大于 0，起始点小于终止点的情况，其余情况可通过这种情况稍加变化而来。

若要绘制一条线，首先要提供这条线的起点和终点，(x1,y1)和(x2,y2)，这两个点可以直接绘制出来，接着绘制两点之前其它的点。由两点可知，这条直线的斜率是k=y2−y1x2−x1（看到这里出现的浮点数与乘除法不要着急，后续会化简让它消失）。

斜率k的值意味着，x=x+1的时候，y=y+k。那么这个增加后的y，它肯定与某单个像素的纵轴有一个交点，接下来让这个交点与这单个像素纵轴的中点（0.5）进行比较，大于等于中点，就规定这个点的实际绘制y值加 1，否则不增加。反复执行上一段的逻辑，直到绘制到终点，整条线绘制完毕。

Bresenham是绘制线段最有效率的算法之一,该算法仅仅使用[整型](https://so.csdn.net/so/search?q=%E6%95%B4%E5%9E%8B&spm=1001.2101.3001.7020" \t "https://blog.csdn.net/jinbing_peng/article/details/_blank)加减及字节位操作运算 (省去了用于计算斜率的除法运算)，因而计算效率非常高。

1. **Bezier 曲线及B样条曲线的特性有哪些？如何实现Bezier 曲线及B样条曲线的相互转化？（15分）**

**Bezier曲线的特性：**

（1）全局性：最重要的性质，贝塞尔方法无法对曲线形状进行局部控制，改变任一控制点位置时，整个曲线均受到影响。

（2）端点性：曲线只经过两个端点的控制点（起点和终点），其它所有点只是逼近，一般不经过。

（3）起点和终点处的切线方向和特征多边形的第一条边及最后一条边相一致。

（4）几何不变形：曲线的几何特性不随坐标变换而变化，形状仅与控制多边形各顶点的相对位置有关，而与坐标系的选择无关。

（5）贝塞尔曲线的导数还是贝塞尔曲线。

（6）凸包性：凸包就是包含所有顶点的最小凸多边形。凸多边形的任一边延长，其它边都在它的一侧。贝塞尔曲线始终会在包含了所有控制点的最小凸多边形中, 不是按照控制点的顺序围成的最小多边形。通过控制点的凸包来限制规划曲线的范围。

（7）变差缩减性：当特征多边形为平面图形时，平面内任意直线与曲线的交点个数不多于此直线与其特征多边形的交点个数。此性质反映了贝塞尔曲线波动性小于其特征多边形，即比多边形折线更顺滑。

**B样条曲线的特性：**

（1）局部性：和贝塞尔曲线相比最重要的区别。B样条基函数的局部支撑性决定了B样条的局部性。k阶曲线上的一点至多与k个控制点有关，与其它控制点无关。故移动曲线上第i个控制顶点Pi，至多影响到定义在此点对应区间上的那部分曲线的形状，对曲线的其余部分不发生影响。

（2）变差缩减性：同贝塞尔曲线。

（3）几何不变性：同贝塞尔曲线。

（4）凸包性：B样条曲线落在Pi构成的凸包之中，其凸包区域小于或等于同一组控制顶点定义的贝塞尔曲线凸包区域。

（5）不同于贝塞尔曲线，B样条曲线不一定经过两个端点。

贝塞尔曲线是理解B样条的基础。贝塞尔曲线具有全局性和极佳的平滑性。其阶数取决于控制点个数。移动一个控制点，整段曲线都会变化。

B样条在贝塞尔曲线的基础上，延续其优点，弥补其不足。B样条具有局部性和一定程度的平滑性。B样条可以自定义阶次，移动控制点仅仅改变曲线的部分形状。

对于NURBS曲线，首先用节点插入的算法将三次NURBS曲线转化成一组三次有理Bezier曲线，然后再结合有理Bezier曲线形式简单的优点，求解有关B样条的问题，对任意阶均匀B样条，给出任意阶均匀B样条和Bezier曲线之间相互转换矩阵的计算方法。

1. **简述深度缓冲(Z-buffer)算法原理。（10分）**

Z-buffer算法的原理:先将待处理的景物表面上的采样点变换到图像空间,即屏幕坐标系,计算其深度值,并根据采样点在屏幕上的投影位置,将其深度与已存储在Z缓存器中相应像素处的原可见点的深度值进行比较。如果新的采样点的深度(Z值)大于原可见点的尝试表明新的采样点计划遮住了原来的可见点,则用该采样点处的颜色更新帧缓存器中相应像素的颜色,同时用其深度值更新Z缓存器中的深度值;否则不作更改。

该算法需要两个缓存器，一个是用来存放颜色的颜色缓存器，另一个是用来存放深度的深度缓存器。利用深度缓存器，可以进行可见性的判断，消除隐藏对象。其具体做法是：

首先对深度缓存器和颜色缓存器进行初始化，把深度缓存器中所有单元置成一个最大可能的深度值，把颜色缓存器中各单元置成背景颜色。然后将场景中的物体不分次序地[投影](https://baike.baidu.com/item/%E6%8A%95%E5%BD%B1?fromModule=lemma_inlink" \t "https://baike.baidu.com/item/%E6%B7%B1%E5%BA%A6%E7%BC%93%E5%AD%98%E7%AE%97%E6%B3%95/_blank)到象平面上去。对于每个[投影](https://baike.baidu.com/item/%E6%8A%95%E5%BD%B1?fromModule=lemma_inlink" \t "https://baike.baidu.com/item/%E6%B7%B1%E5%BA%A6%E7%BC%93%E5%AD%98%E7%AE%97%E6%B3%95/_blank)点(象素)，把投影物体在该点处的深度与深度缓存器中相应位置上的深度值进行比较，如果前者小于后者，那么就把当前被投影物体的颜色写到颜色缓存器中去，同时用当前投影物体的深度去更新深度缓存器中相应象素的深度，否则不做任何操作。

1. **传统图形学与智能计算机图形学的联系与区别。（10分）**

**联系：**

1.计算机图形学中也会用到 传统图形学，现今的三维游戏为了增加表现力都会叠加全屏的后期特效，原理传统图形学，只是将计算量放在了显卡端。比如近年来的热点——增强现实（AR），它既需要计算机图形学，又需要 传统图形学。

2.得益于各个领域的共同进步，所以能看到计算机图形学和传统图形学现在出现越来越多的交集。如果根据观测量，传统图形学可以越来越准确的估计出越来越多的变量，那么这些变量套到计算机图形学算法中，就可以模拟出一个跟真实环境一样的场景出来。

与此同时，计算机图形学需要构建更真实的场景，也希望能够将变量更加接机与实际，或者通过算法估计出来，这就引入了图形学的动机。这也是近年来三维重建算法，同时大量发表在图形学的会议的原因。随着CV从2D向3D发展，以后两者的交集会越来越大。

3.计算机图形学与传统图形学虽然有较大区别，但是他们之间在特定条件下也可以实现转化，一种转化是对象和输入、输出设备之间的硬转化，另外一种是对象文件格式之间的软转化。

**区别：**

1.计算机图形学和传统图形学是同一过程的两个方向。计算机图形学将抽象的语义信息转化成图像，传统图形学从图像中提取抽象的语义信息。

2.计算机图形学是一种使用数学算法将二维或三维图形转化为计算机显示器的栅格形式的科学；传统图形学图像处理泛指如何对各类图形或图像进行各种处理；计算机图形学偏向于研究使用计算机技术处理图形的一些基础方法，而；传统图形学图像处理则是各种处理的统称。

计算机图形学主要研究的是模型在三维空间中的数学表达，如投影矩阵之类的，很理论纯数学。而；传统图形学图像处理是一种应用类的学科，需要简单的工科基础。

3.计算机图形学是研究怎样利用计算机来显示、生成和处理图形的原理、方法、和技术的一门学科。计算机图形学试图将参数形式的数据描述转换生成（逼真的）图像。；传统图形学图像处理则着重强调图像之间进行变换，它旨在对图像进行各种加工以改善图像的视觉效果，计算机视觉是研究用计算机来模拟生物外显或宏观视觉功能的科学和技术，它模拟对客观事物模式的识别过程，是从图像到特征数据对象的描述表达处理过程。

4.计算机图形学，输入的是对虚拟场景的描述，通常为多边形数组，而每个多边形由三个顶点组成，每个顶点包括三维坐标、贴图坐标、rgb 颜色等。输出的是图像，即二维像素数组。

而传统图形学中，计算机视觉，输入的是图像或图像序列，通常来自相机、摄像头或视频文件。输出的是对于图像序列对应的真实世界的理解，比如检测人脸、识别车牌。数字图像处理，输入的是图像，输出的也是图像。Photoshop 中对一幅图像应用滤镜就是典型的一种图像处理。常见操作有模糊、灰度化、增强对比度等。

1. **PointNet是三维点云学习方面的一个开拓性工作。简要介绍PointNet （CVPR2017）的思路。（15）**

PointNet，由斯坦福大学的Charles R. Qi等人在「CVPR2017」上发表，是一个「端对端」的神经网络，可以直接将「点云数据」作为输入，通过学习，实现对3D点云数据目标的「分类与分割」。

[点云](https://so.csdn.net/so/search?q=%E7%82%B9%E4%BA%91&spm=1001.2101.3001.7020" \t "https://blog.csdn.net/weixin_44378835/article/details/_blank)是不规则的，大多数研究人员将这些数据转换为规则的三维体素网格或图像集合。然而，这会使数据冗余。出于这个原因，PointNet研究团队专注于使用简单点云的三维几何体的不同输入表示，并将生成的深层网络命名为PointNet。 输入点云是简单而统一的结构，可以避免网格的组合不规则性和复杂性的情况，因此更容易学习。PointNet是一种直接处理点云的新型神经网络，它很好地体现了输入点云的序列不变性，从对象分类，部分分割到场景语义分析等方面提供了一个完整的体系结构。

PointNet 是一个统一的体系结构，它直接将点云作为输入，并为输入的每个点输出整个输入的每个分类标签或每个点分段/每个部分标签。其网络的基本架构非常简单，因为在初始阶段，每个点都被相同和独立地处理。在基本设置中，每个点仅由其三个坐标（x， y， z） 表示，可以通过计算法线和其他本地或全局特征来添加其他维度。

PointNet模型的输入是包含n个点的三维点云(x, y, z) , 原始数据通过一个「3D 空间变换矩阵预测网络」 T-Net(3)，估计出3x3的变换矩阵T(3) 并作用在原始数据上，实现数据的对齐。对齐后的数据会以点为单位，通过一个共享参数的双层「感知机模型mlp」进行特征提取。每个点提取出64维的特征，再通过特征空间变换矩阵预测网络 T-Net(64) 预测64x64的变换矩阵，作用到特征上，实现对特征的对齐。然后继续利用三层感知机（64,128,1024）进行以特征点为单位的特征提取，直到把特征的维度变为「1024」，继而在特征空间的维度上进行Max Pooling，提取出点云的「全局特征向量」。

一方面，全局特征向量经过三层感知机模型，得到k分类输出，实现k分类任务。

另一方面，全局特征向量与前面的64维特征向量concat，通过三层感知机模型得到128维的点特征「point features」，再经过两层感知机模型，得到n\*m的特征向量。n表示输入点的个数，m表示每个点所属的类别信息，从而完成分割任务。

在PointNet模型中主要用到的的关键方法有：

1.无序输入的对称函数。

为了使模型对输入排列保持不变性，有三种策略：

将输入按规范顺序排序；

将输入视为训练RNN的序列，但通过各种置换来扩充训练数据；

使用简单的对称函数来聚合每个点的信息。这里，一个对称函数以n个向量作为输入，并输出一个新的向量，该向量与输入顺序不变。用一个多层感知器网络来近似h，用一个单变量函数和一个最大池函数的组合来近似g。实验证明这种方法很有效。通过h的集合，我们可以学习一些f来捕捉集合的不同性质。

2.局部和全局信息融合。

点分割需要结合局部和全局知识。具体做法是：在计算出全局点云特征向量后，将全局特征与每个点特征连接起来，然后，基于合并的点特征通过3层mlp提取新的逐点特征，提取出的每点特征能同时感知局部和全局信息。

3.连结对齐网络。

如果点云经过一定的几何变换，如刚性变换，则点云的语义标注必须是不变的。因此，我们期望由我们的点集学习的表示对这些变换是不变的。我们通过一个小型网络「T-Net」预测仿射变换矩阵，并将此变换直接应用于输入点的坐标。小网络本身类似于大网络，由点无关特征提取、最大池和全连通层等基本模块组成。

这种思想还可以进一步扩展到特征空间的对齐。我们可以在点特征上插入另一个对齐网络，并预测一个特征转换矩阵来对齐来自不同输入点云的特征。然而，特征空间中的变换矩阵维数远高于空间变换矩阵，这大大增加了优化的难度。因此，我们在softmax训练损失中添加了一个正则化项。我们将特征变换矩阵约束为接近正交矩阵,正交变换不会丢失输入中的信息，因此是需要的。我们发现通过加入正则化项，优化变得更加稳定，我们的模型获得了更好的性能。