

**毕业论文**

自动化纹理图集的设计与实现

**学生姓名**： **学号**：

刘璞

162052115

计算机工程系

**系 部：**

数字媒体技术

**专 业：**

孔令德（教授）

**指导教师：**

二○一八年六月

诚信声明

本人郑重声明：本设计及其研究工作是本人在指导教师的指导下独立完成的，在完成设计时所利用的一切资料均已在参考文献中列出。

本人签名： 年 月 日

自动化纹理图集的设计与实现

摘要

20世纪90年代，计算机图形学技术异军突起，特别是GPU技术的发明和普及，使得CG电影工业和电子游戏产业蓬勃发展。在GPU中，曲面的数据处理可以分为两条流水线：一条处理曲面的几何数据和几何变换，如曲面的平移旋转、射影变换等；另一条处理曲面的纹理信息，如颜色、法向量场、局部几何细节鳞片结构、局部材质特性BRDF等。几何数据结构一般是多面体三角网格，纹理数据结构一般是平面图像。在图形渲染过程中，两条流水线的处理结果相互融合，将平面的二维纹理粘贴到弯曲的三维曲面上，这一技术称为曲面纹理贴图。

纹理贴图技术在数学上等价于求从曲面到平面区域的一个光滑双射，这称为所谓的曲面参数化问题。如图1-1所示，我们将各种曲面映射到平面区域，然后在平面参数区域贴上纹理图像，通过逆映射将纹理图像拉回三维曲面。如图1-2所示，我们将大理石纹理贴在米开朗基罗的大卫头像上面，从而获得了大理石雕像的质感。

关键词：曲面纹理贴图，

**Design and implementation of NURBS surface subdivision modeling**

**abstract**

NURBS is a non-uniform Rational b-spline (non-uniform Rational b-spline), which is the most commonly used modeling method in geometric modeling. NURBS has many good mathematical properties, including the nature of node insertion segmentation to be studied in this paper. The principle of surface subdivision technology is to subdivide the surface several times according to a set of subdivision rules, so as to subdivide the surface of the model to a certain degree and make the model look smooth and delicate enough.

Surface subdivision can generate multiple vertices based on the original model vertex data, and the surface formed by new vertices and original vertices still retains its geometric properties. These vertices are not virtual, they are real vertices. Unlike the texture mapping in computer graphics, which simulates real images, the segmented image effects are the same as those directly designed when modeling. However, data volume and processing time are also relatively increased. With the development of computer hardware technology, real time will be better and better. The segmentation algorithm has been continuously developed, including catmull-clark segmentation, doo-sabin segmentation, Loop segmentation, butterfly segmentation, NURBS node insertion segmentation, t-spline segmentation and so on.

In this paper, through research on segmentation methods above, item by item, from the perspective of the NURBS node insert segmentation algorithm, through the establishment of a small NURBS modeling system, the program can realized the NURBS node insert segmentation algorithm, using NURBS subdivision modeling mapped the relevant technology of three-dimensional objects.

**Key words:** NURBS, node insertion, NURBS modeling system, subdivision.

目录

[1 绪论 1](#_Toc516061417)

[1.1引言 1](#_Toc516061418)

[1.2计算机辅助设计中建模技术的发展 1](#_Toc516061419)

[1.2.1 计算机辅助设计研究对象及核心问题 1](#_Toc516061420)

[1.2.2 CAGD中形状数学描述的要求 2](#_Toc516061421)

[1.2.3 NURBS的发展 3](#_Toc516061422)

[1.3选题的背景及内容概要 4](#_Toc516061423)

[1.3.1选题的背景 4](#_Toc516061424)

[1.3.2选题内容概要 5](#_Toc516061425)

[2.1 B样条的知识 6](#_Toc516061426)

[2.1.1 B样条定义 6](#_Toc516061427)

[2.2 NURBS知识 7](#_Toc516061428)

[2.2.1 NURBS方法的提出 7](#_Toc516061429)

[2.2.2 NURBS曲线方程的有理分式表示 8](#_Toc516061430)

[2.2.3 NURBS曲面方程的有理分式表示 8](#_Toc516061431)

[3 细分的相关知识 10](#_Toc516061432)

[3.1 几种细分的方案 10](#_Toc516061433)

[3.1.1 Catmull-Clark双三次B样条细分 10](#_Toc516061434)

[3.1.2 Loop细分 11](#_Toc516061435)

[3.1.3 蝶形细分 13](#_Toc516061436)

[3.1.5 T样条细分 14](#_Toc516061437)

[3.2 本文的细分方法 14](#_Toc516061438)

[3.2.1细分的方法 15](#_Toc516061439)

[3.2.2细分算法的实现 15](#_Toc516061440)

[4 细分建模技术设计实现 18](#_Toc516061441)

[4.1 整体架构设计 18](#_Toc516061442)

[4.2模块功能设计 18](#_Toc516061443)

[4.3 NURBS建模系统 19](#_Toc516061444)

[4.3.1系统主要功能界面 19](#_Toc516061445)

[4.4 基于Qt平台的设计 22](#_Toc516061446)

[4.4.1主要的函数介绍 22](#_Toc516061447)

[4.5 主要的类说明 25](#_Toc516061448)

[5 结论 26](#_Toc516061449)

[5.1已完成的工作 26](#_Toc516061450)

[5.2今后进行的工作 26](#_Toc516061451)

[参考文献 27](#_Toc516061452)

[致谢 29](#_Toc516061453)

**1 绪论**

**1.1引言**

计算机辅助几何设计（Computer Aided Geometric Design，CAGD）是一门涉及代数几何、计算数学、微分几何、计算数学、交换代数、函数逼近论及计算科学等的新兴学科。CAGD研究的主要内容是“在计算机中进行曲线曲面的表示和逼近”，或者称之为“产品形状信息如何在计算机中得到有效的表示、处理和分析”，1974年在美国的Utah大学举行了著名的CAGD学术会议，这次会议的举办使CAGD正式作为计算数学的一个分支学科正式建立[1]。CAGD学科正式建立后，计算机辅助几何设计经过不断的发展，在计算机设计和制造(CAD/CAM）的数学理论和几何体的构造方面有显著的成果，体现在现代飞机、 船舶、 汽车设计、工程器件模具设计、生物医学图像处理等方面的广泛应用。不仅如此，它还扩展到许多新领域，包括计算机动画、生物工程分析、建筑美学设计、工业设计与制造、机器人学、虚拟服装设计、地质与气象测绘和科学可视化。计算机辅助几何设计的产生和发展，不仅是现代工业的需要，而且对促进现代CAGD的发展起着重要的作用。

**1.2计算机辅助设计中建模技术的发展**

**1.2.1 计算机辅助设计研究对象及核心问题**

作为一门随着航空、汽车等现代工业发展与计算机出现而产生与发展起来的新兴学科，计算机辅助几何设计经过不断发展，研究的对象尽管已经扩展到了四维曲面的表示与显示等，但其主要的研究对象仍是工业产品的几何形状方面，工业的产品形状总体上可分为两大类：一类是仅由基本初等解析曲面（比如像平面、圆柱面、圆锥面、球面、圆环面等）所组成，机械零件总体是属于这一类，这一类曲面可以用画法几何与机械制图方法完全清楚地表达和传递所包含的产品的全部形状信息[13]；第二类是不能由基本初等解析曲线曲面所组成的，须得以复杂自由变化的曲线曲面所组成，也就是所谓的自由型曲线曲面组成，比如像飞机、汽车、船体形状部分[13]。显然，后一类形状简单使用画法几何与机械制图方式是达不到这种工业级别要求的。随着计算机出现和计算机软硬件快速发展，采用数学方法定义自由型曲线曲面方法不再受到限制，并在现实得到实际应用，这更加促成了CAGD产生与发展。根据所定义的几何信息和所提供的方法，建立了相应的曲线和曲面方程，即数学横形，并在计算机上处理曲线和曲面上大量的点和其他信息。在这一过程中，通过分析和综合之后就可了解所定义形状所具有的局部和整体的几何特征信息，且实时的显示与交互的修改工作到达了同步进行。形状的几何定义为所有后处理（如数值控制、物理特性计算、有限元分析等）提供必要的先决条件。

形状信息在计算机的表示、分析和综合中，其核心问题是形状的信息在计算机中的表示，也就是不仅要找到一种适合计算机的表示和处理，而且足够满足形状表示与几何设计的要求，同时也得是能便于形状信息传递和产品数据交换的形状信息描述的数学方法[13]。

**1.2.2 CAGD中形状数学描述的要求**

随着CAGD学科及其应用的不断发展，人们对于形状数学描述要求也不断的加深。

要在计算机内表示某一工业产品的形状，要求其数学的描述应尽可能多的保留产品形状的性质。从实现计算机对形状的处理、便于形状信息的传递与产品数据的交换角度来看，施法中指出应满足下列的要求[13]。

1. 具有唯一性：自由型曲线曲面在传统上采用模线样板的方法获取形状数据源，但不能保证形状的定义唯一性，才转而采用数学的描述，可见唯一性是对形状的数学描述当中首项的要求，唯一性对所采用的数学方法要求是，由已给定有限信息所决定形状应得是唯一的[13]。

2. 具有几何不变性：当使用有限信息确定形状时，确定顶点的相对位置，确定的形状是固定的，并且不随坐标系的变化而变化，标量函数并不具几何不变的特性，参数曲线曲面在一定的条件下具有几何不变的特性[13]。

3. 具有容易定界性：产品形状是有边界的，形状的数学描述应该得容易定边界[13]。

4. 具有统一性：能够统一的表示各种形状、处理各种形状，包括各种各样特殊情况，对统一性的更高一级的要求是能找到统一数学形式，也就是不仅表示各种自由型曲线曲面的产品，而且得能表示初等的解析曲线曲面产品，达到可建立统一数据库，方便进行形状信息的传递及产品数据的交换[13]。

5. 应便于计算处理：方便在计算机上的实现和在计算机上推广进行应用[13]。

在形状的表示与设计角度考虑，形状的数学描述得满足下列硬性要求：

1. 应具有丰富的表达力及灵活的响应能力：在形状表示中代表两个基本形状和它们的复合形状，在形状的设计上，设计员要求形状应该是不受限制的，即设计员会将形状频繁的变换，形状数学的描述必须得能适应设计人员绘制任意形状的能力[13]。

2. 应容易实现连接：在很多场合要求形状的光滑连接，通常一个曲线段或者曲面片不足于表达复杂的形状，必须将曲线段经过某种规则连接在一起作为组合曲线，将曲面片经过某种规则拼接成为组合曲面，才能达到要求，足够去描述复杂的形状要求，在设计一条光滑曲线或者是一张光滑的曲面片时，要求曲线段之间、曲面片之间的连接是光滑过度的，形状数学的描述得容易实现这种光滑连接[13]。

3. 易于实现对形状的控制：设计员要求能随心所欲的控制显示的曲线或者是曲面的形状，即形状数学的描述应同时具有整体控制和局部控制的能力，且需要形状数学的描述具有灵活控制性，最后还希望在对形状进行控制的同时能估测会发生的形状变化[13]。

4. 应具有几何直观性：在几何方面应该意义清晰，从几何直观的方面上处理问题通常会比从代数问题的方面更为被工程应用人员所接受，使得其具有更强的生命力[13]。

简单地说，NURBS建模总是由曲线和曲面来定义的，因此在NURBS曲面上生成具有边缘和角点的边缘是非常困难的。正因为如此，我们可以利用它来制作各种复杂的曲面造型和特殊效果，如人体皮肤、外观或流线型跑车。

**1.2.3 NURBS的发展**

在20世纪80年代中期以后， NURBS良好的发展前景被国际上所认可，更是在1991年被国际标准化组织ISO为关于工业产品几何定义的STEP，把NURBS方法作为定义产品形状的唯一数学方法。在1992年，国际标准化组织将NURBS加入到设备无关性图形交互显示的编程接口，在国际标准的PHIGS中，将NURBS作为PHIGS Plus的扩充部分。同时还把Bezier、有理Bezier、均匀B样条和非均匀B样条都统一加到NURBS当中。

NURBS曲线与NURBS曲面技术是为使用计算机进行三维的建模要求而建立起来的。通常三维建模过程中使用曲线与曲面来表现产品的轮廓和外形。NURBS曲线与NURBS曲面是用数学表达式来构建的，而且NURBS的数学表达式是一种组合体。

NURBS全称为Non-Uniform Rational B-Splines，顾名思义即是非均匀有理B样条的意思。具体的解释如下：

Non-Uniform（非均匀性）：是指一个控制顶点的影响区域范围能够进行控制。这一性质在遇到一些不规则曲面片的时候具有重要的意义。

Rational（有理）：是指每个由NURBS构建的物体都可以使用有理多项式形式所表达定义。

B-Spline（B样条)：是指用足够多的线段来构建一条曲线，即通过对一原始线段不断的插值逼近曲线。

**1.3选题的背景及内容概要**

**1.3.1选题的背景**

NURBS曲面和其他曲面都是在de Boor三次样条曲面插值的研究基础上，加入了张量积曲面定义，其在构造CAGD的新式曲面中很常见。在CAGD发展的初期研究的曲面主要在舰船、飞机和汽车的外形设计上，它们的形状都有共同的特点，即形状都是矩形域的，因此CAGD开始研究矩形区域类的曲面（NURBS是属于为矩形区域类的）。随着越来越多复杂产品形状的出现，CAGD应用范围也不断的扩大，单个矩形域已不能满足设计者实际的建模要求，特别是在CG动画制作的领域。前期的曲面拼接技术与裁剪技术在复杂CG模型中的构建是不容易实现的。使用表面切割技术不仅代价高，而且结果不准确。拼接时更是难于保证表面的平滑过渡连接，裁剪表面之间拼接的光滑过渡性也更加难以得到很好控制。因此，CG动画的几何建模一直受到曲面裁剪与拼接问题的困扰。直到人们提出了细分曲面。细分曲面不需要切割、不需要关节连接的特性成功的解决了曲面裁剪与拼接的问题。它可以在任意拓扑网格上便捷的构造光滑的表面。

细分曲面突破张量积曲面的四边形矩形域约束。数值计算稳定，算法简单，可部分细分。与NURBS曲面相比，它更适合于CG动画和雕刻曲面设计。细分曲面的原理是给定了一个初始的多边形网格信息，并通过定义一套细分规则不断在已有网格上插入新的控制顶点，然后将多边形网格多次进行细化，最后得到了一张具有光滑连续性的曲面。按照此方法，定义一个特殊的局部细分规则，并应用在不规则的拓扑网格上，不通过表面的裁剪与拼接就可以构造出具有尖锐特征的表面。最早的细分算法是图形艺术家Chaikin在1974年CAGD国际会议上提出用于快速生成一条光滑的曲线的多边形割角算法[2]。在1978年Catmull 和Clark 提出了CatmullClark细分曲面，是对任意拓扑网格上的三阶齐次B样条曲面的扩展[3]。同年Doo 和Sabin 也提出了Doo Sabin细分曲面，是对任意拓扑网格上双均匀B样条曲面的扩展[4]。这两种细分曲面的提出标志着这一新曲面形式的出现。1987年，Loop将四次三项样条推广到任意的三角网格上，构建出了基于三角网格的Loop细分曲面[5]。在三角网格方面，Dyn和Levine则利用四点插值的细分方法，构建了基于三角网格插值的蝶形曲面(Buterfly surface)[6]。1998年，Sederberg、Zheng和Sewell等人将节点距方法引入细分曲面构建中，获得了两种非均匀有理细分曲面-广义的CatmullClark细分曲面（NURCCs）和广义的Doo-Sabin细分曲面[7]。在2003年Sederberg、Zheng和Bakenov等人利用T控制点改善了NURCCs，在T-NURCCs中实现局部插入T控制点方法[8]。Stam给出了Catmull-Clark细分曲面精确的计算方法，实现通过细分曲面可以精确计算为任意参数的参数曲面 [9]。

细分曲面理论在学术界不断的深化与完善，细分曲面作为CG动画制作的一种有效的曲面造型方法也在商业界不断被使用。许多三维建模系统，如开源软件Brand、Pixar公司的Renderman和FEDES Effice公司的胡迪尼，已经在其建模系统中添加了细分曲面技术。细分曲面技术像产品设计中的工业标准一样，已经成为CG动画制作的行业标准。

**1.3.2选题内容概要**

本文在分析国内外关于曲线曲面细分建模技术的基础上，结合先前的研究，采用了节点插入来进行NURBS曲线曲面细分，并设计一个NURBS曲面细分建模的小型系统，将细分技术应用在建模系统上，实现三维的曲面细分建模。从NURBS细分原理到代码的实现，使更多的人学会NURBS和NURBS细分，并对NURBS能获取更加深入的了解，并能扩充NURBS的新功能，让NURBS发挥出更强大的作用。

**2 NURBS相关的知识**

NURBS是非均匀有理样条的缩写。要理解NURBS曲线和曲面的细分，首先要了解B样条和NURBS的相关知识。

**2.1 B样条的知识**

1946年Schoenberg就提出了B样条的理论[10]，但直到1967年关于B样条方面的论文才正式被发表出来。1972年， de Boor [11]和Cox[12]分别给出关于B样条的标准算法。

**2.1.1 B样条定义**

1. B样条曲线公式：



(2.1)



公式(2.1)中，是多边形的控制顶点。将控制顶点连接起来的多边形被称为B样条控制多边形，简称为控制多边形。k表示样条的次数，被称为B样条基函数。非递减参数组成的序列决定了B样条的基函数。舍恩伯格研究发现B样条具有局部支撑的特性。



1. B样条的递推定义：

经过人们不断的研究，B样条具有多种等价定义方式。其中，de Boor-Cox递推公式，是由de Boor-Cox提出的B样条定义方式，这种定义方式被作为标准算法的递推定义方式。这个递推公式的发现是B样条理论最重要进展之一。经过人们不断的改进，现在的de Boor-Cox递推公式为：



(2.2)

公式(2.2)中，的双下标中第一标号i表示基函数和控制顶点的序号，第二标号k表示B样条的次数。根据德布尔-考克斯的递推公式可知，要确定第i个k次B样条的基函数，就得需要使用共k+2个节点，即。人们将区间称作基函数的支撑区间。基函数的第一标号i等于支撑区间左端节点的标号i，表示为此段B样条在参数u序列上的位置。B样条曲线方程中的n+1个控制顶点需要使用到共n+1个k次B样条基函数。B样条曲线中的所有支撑区间等于B样条基函数的节点矢量。



1. B样条的曲面方程：

定义一张个控制顶点的控制网格。同时给出参数u和v的次数分别为和，两个节点矢量U和V分别为、。则其次张量积样条曲面方程表示为



(2.3)

公式(2.3)中，B样条与分别由公式(2.2)所决定。

**2.2 NURBS知识**

**2.2.1 NURBS方法的提出**

在飞机和绝大多数机械零件形状的设计中，一些二次曲线弧和二次曲面通过采用有理贝齐尔曲线和曲面就可以获得到精确的描述。但有一些特殊的形状如叶轮、飞机机身的表面等由横向框架和自由样条曲线相组合、混合的样条曲线，就不能获得到精确的描述。

B样条法在自由曲线曲面的表达和设计方面具有很大的应用价值。然而，表达和设计由两个表面和平面构成的基本表面是困难的。B样条曲面包括Bezier曲面的特殊情况，除了抛物面之外，不能准确地表示两个曲面，而只能给出。近似表示法。但近似的表示将带来后面处理上的麻烦，使本来简单的问题变得复杂化，还会出现原来不存在的设计与计算的误差问题。例如，为用Bezier方法较精确地表示某一半径的半圆，需要用到五次Bezier曲线，并必须专门计算其控制顶点[13]。如果改变圆的半径或要求更高的精度，就必须重新确定次数及计算控制顶点。显然，专门用于自由型曲线曲面的B样条方法包括Bezier方法已不能适应初等曲面的要求。为了准确地表达二度曲线和二度曲面，我们必须使用另一组数学描述方法，如隐式方程。但使用隐方程不仅会带来隐方程表示所存在的问题，而且将导致一个几何设计系统得使用两套不同数学方法。这将使系统变得十分复杂，这是系统设计人员最不想看到的。解决这一问题的方法显然是对现有的B样条方法进行变换，并保留其B样条的性质，并使其能精准表示二阶曲线和二阶曲面。因此，人们提出了非均匀有理B样条方法，简称NURBS。

**2.2.2 NURBS曲线方程­­­­的有理分式表示**

一条k次NURBS曲线表示为一分段有理多项式矢函数

(2.4)



公式（2.4）是NURBS曲线的数学定义。式中被称为权重或权重因子（权重），它们分别与控制点相对应，一个控制顶点对应着一个权因子。首末权因子要求的大于0，即，其余大于等于，即0。同时要求顺序k个权因子不能同时为零，以防因分母为零使曲线退化为一点。被称为控制顶点，其连接起来组成NURBS的控制多边形。是k次规范B样条基函数，和B样条一样，是由节点矢量按de Boor-Cox递推公式决定的。假如将NURBS开曲线节点矢量两端节点的重复度取为k+1，即，则NURBS就会变成Bezier曲线。在大多数实际应用中，终端点的值分别为0和1。因此，曲线定义域常取为。



**2.2.3 NURBS曲面方程的有理分式表示­­­­**

在U和V方向使用 NURBS曲线的方法，就构造出了一张NURBS曲面，一张次NURBS曲面的有理分式表示



(2.5)

公式（2.5）中是控制权因子，规定四角顶点处权因子等于0，即、、、，其余大于等于0，即且顺序 个权因子要求不能同时为零。控制顶点按矩阵排列，形成一个的控制网格。和B样条一样，一个顶点对应一个控制权因子。NURBS曲面的和分别为u向k次和v向 次的规范B样条基。它们分别由u向和v向的节点矢量与按de Boor递推公式决定。



**3 细分的相关知识**

**3.1 几种细分的方案**

**3.1.1 Catmull-Clark双三次B样条细分**

Catmull-Clark的细分算法是基于双三次B样条的计算机图形细分建模技术，是一种光滑表面细分曲面建模的算法。它由Edwin Catmull和Jim Clark在1978 年设计，并将双三次均匀B样条曲面推广到任意拓扑结构层面上[4]。

Catmull-Clark表面是递归细分的，他们使用细化方案如下：

从任意多面体的网格开始，当前网格中的所有顶点都称为原顶点。

● 在多面体的每个面里面添加一个新的顶点；

● 将每个面中新添加顶点设置为其面中所有原顶点的平均值的位置；

● 在多面体每个面的边界上添加一个新的顶点；

● 将每个边界新添加的顶点设置为两个相邻面点及其两个原始顶点的平均值。

● 对于每个面，将在每个面中添加的顶点连接到每个边界顶点上；

新的多面体网格比旧网格很具有光滑性, 并且新的网格只包含四边形。

通过Catmull-Clark细分的程度是使其表面获得一阶几何连续性或者更高的二阶几何连续性。

Catmull-Clark细分方法创建出的形状并不是由数学推导来选择任意形状的方法，尽管Catmull和Clark非常努力地证明这个细分方法收敛于双三次B样条曲面。

对新增的顶点位置及其原顶点位置的更新规则如下图：

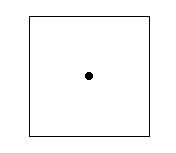
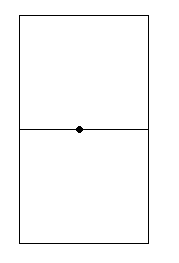




图3.1 网格内部F-顶点位置

图3.2网格内部E-顶点位置





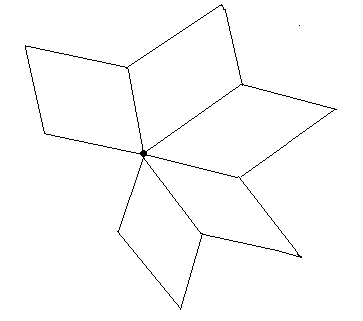


图3.3 网格边界E-顶点位置





图3.5 网格内部V-顶点位置

图3.4 网格边界V-顶点位置

1. 网格内部F-顶点的位置：

设四边形四个顶点为，则新增加顶点的位置为。



(3.1)



1. 网格内部V-顶点的位置：

设内部顶点v0相邻的点为，则将顶点更新到位置，其中α、β和γ分别为α = 1 - β - γ。



1. 网格边界V-顶点的位置：

设边界顶点v0两个相邻的点为v1、v2，则该顶点新的位置为。



(3.2)

1. 网格内部E-顶点的位置：

　设内部边界的两个端点为v0、v1，并且与边缘相邻的两个四边形顶点是和，则新添加的顶点位置是。



(3.3)

1. 网格边界E-顶点的位置：

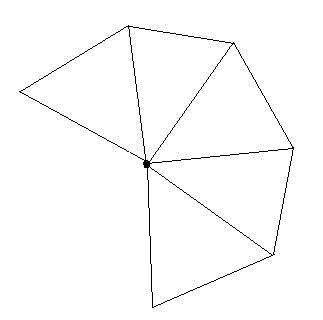
　设网格边界的两个端点为v0、v1，则新增加顶点的位置为



(3.4)

**3.1.2 Loop细分**

Loop细分是由Charles Loop于1987年为三角网格开发的近似细分规则[14]。它被划分成1-4个三角形，每个边被计算来生成一个新的顶点，并且每个原始顶点被更新。下面是Loop环细分方案的细分掩模。更新顶点位置和原始顶点位置的新规则如下：





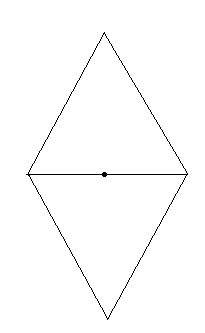


图3.6网格内部E-顶点的位置

图3.7网格内部V-顶点的位置







图3.8网格边界E-顶点的位置

图3.9 网格边界V-顶点的位置

1. 网格内部V-顶点的位置：

设网格内顶点v0相邻的点分别为，则顶点新的位置为



(3.5)



公式（3.5）中 。

1. 网格边界V-顶点的位置：

设边界顶点v0两个相邻的点为v1、v2，则顶点新的位置为



。 (3.6)

1. 网格内部E-顶点的位置：

设内部边上的两个端点分别表示为v0、v1，对应的另两个顶点分别为v2、v3，则新增加顶点的位置为 。



1. 网格边界E-顶点的位置：

设边界上的两个端点分别表示为v0、v1，则新增加顶点的位置表示为



(3.7)

**3.1.3 蝶形细分**

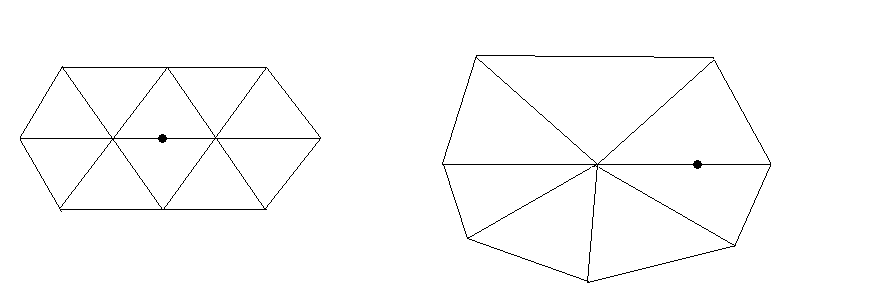
Dyn、Gregory和Levin首先提出了蝶形插值细分的算法[15]，使用细分的规则如图(3.10) 所示。细分算法可以在规则区域获得光顺次序，但是边缘的平滑度降低（连接到一个点的边缘的数量被称为奇点附近的点的人类程度。如果一个内点的人度不为6或者一个边缘点的人度不为4。则称这样的点为奇异点，否则称为规则点[16])。Denis Zorin改进了Dyn、Gregory和Levin的细分规则[17]，他在奇异点周围定义了新细分的规则（如图3.10所示），这种新细分的规则使得细分面片在奇异点周围也具有光滑的连续性。同时，Zorin在边缘定义了一种新的细分规则，以保证细分曲面在边缘上具有良好的形状。尽管Zorin解决了传统螺形细分法在奇异点处光滑连续性很差的问题，但是他的算法在用于处理一般的三维网格的时候，细分出来的面片效果很差，达不到设计人员的要求，究其原因是细分面片的光顺性很差，进而严重影响了细分面片的视觉效果。



图3.10 细分规则

细分曲面是通过对初始给定的网格进行无限细分次数后得到的近似结果。但在在实际的运用中，进行无限次数的细分是不现实。因此，人们通常是在细分网格的尺寸小于屏幕的分辨率时，对于初始网格只执行两个细分，并且在检测到两个细分之后，将补丁的近误差和平滑的水果用作初始调整的基础。细分次数越多，效果越好，但计算量越大，光顺效率越低。因此，这两个细分是效率和效率之间的权衡。

**3.1.4 Doo-Sabin 细分**

Doo-Sabin细分算法是二次均匀B样条曲面二分技术的推广，它由Daniel Doo和Malcolm Sabin于1978年开发[18][14]。于Loop细分算法相比，Doo-Sabin细分算法的控制网格是四边形的细分算法。

对于每一个控制网格的面F，设其顶点用下

列公式计算

(3.8)

其中



(3.9)

**3.1.5 T样条细分**

Sederberg等人提出了T样条的局部细分算法[19]。

T样条局部细分是指在T网格插入一个或多个控制点而不改变T样条的曲面形状。这一方法被称为局部节点插入，因为插入一个控制点到T网格中必须伴随着将节点插入到相邻的混合函数中。

T样条细分算法包含了拓扑和几何这两个阶段。其中，拓扑阶段会指明除了要求插入的控制点之外还有哪些控制点（如果有的话）必须被插入。

在T样条细分过程中，主要会出现以下3种主要Violation情形：

● Violation1一个混合函数根据中的Rule1缺少当前T网格的一个顶点；

● Violation2一个混合函数含有一个当前T网格中的Rulel无法得到的顶点；

● Violation3一个控制顶点没有与之相关联的混合函数。

T样条细分算法的拓扑阶段包括了以下的几个步骤：

1. 插入所有要求的控制顶点到T网格中；
2. 如果有任何的混合函数违反了Violation 1，那么插入需要的节点到这个混合函数中；
3. 如果有任何的混合函数违反了Violation2，那么添加合适的控制顶点到T网格中；
4. 重复第2和3的步骤，直到不再遇到Violation的情形。

**3.2 本文的细分方法**

结合之前对细分建模的研究和细分技术实现的工作，本文选择NURBS插入节点技术作为实现细分建模的研究工作。

**3.2.1细分的方法**

插人节点技术是NURBS基本几何算法中最重要算法之一。其不仅具有重要的理论价值，而且在曲线曲面的设计中有着广泛用途。通过插人节点同时还可以实现对曲线进行分段，便于设计人员实现复杂的建模。

**3.2.2细分算法的实现**

1. NURBS曲线插入一个节点

设已给一条k次NURBS样条曲线



(3.10)

式（3.10）中，在前面NURBS的知识介绍，我们已经知道NURBS样条可以说由节点矢量完全决定。现在我们希望在NURBS的曲线中插入一个节点，即在NURBS曲线定义域的某段节点区间插入一个节点，则新节点的矢量为



(3.11)

重新进行编号后变为

(3.12)

新的一组节点矢量序列决定了新的一组。NURBS样条曲线则由这一组新的和计算出来的新顶点共同表示，公式表示为



(3.13)



控制顶点增加了一个，曲线形状及连续性均保持不变。

1980年伯姆在他节点插入算法给出了新顶点计算公式[20]。

新顶点的计算公式如下：

(3.14)

公式（3.14）中r参数表示为将要插入的节点u在原节点矢量U中已经存在的个数，假如，则。假如已经存在的个数为正整数，则



1. NURBS曲线重复插入同一个节点

有时候会在节点矢量中重复插入次同一节点u，也就是在节点矢量中插入重节点。而将要插入的节点u在原节点矢量U中已经存在的个数为r，也就是已有要求。这时新的节点矢量为



(3.15)

新的一组表示为。新的NURBS曲线表示为



(3.16)

1. NURBS曲面插入节点

一张次的NURBS曲面表示为



(3.17)

公式（3.17）中，NURBS样条曲面由节点矢量和完全决定。在曲面u向节点定义域的某个节点区间内插入一个节点，得到一个新的节点矢量。

在节点区间插入：

(3.18)



新的节点矢量：

(3.19)

重新编号后成为

(3.20)

在曲面v向节点定义域的某个节点区间内插入一个节点，得到一个新的节点矢量。

在节点区间插入：

(3.21)



新的节点矢量：

(3.22)

重新编号后成为

(3.23)

新生成的节点矢量与分别决定了新的与。新的NURBS样条曲面使用这两组新的和，再加上通过计算所得的新顶点与，新的NURBS曲面表示如下



(3.24)

NURBS曲面新顶点位置和NURBS曲线的算法相同，分u向和v向分别计算即可。NURBS曲面中在同一位置重复插入同一节点的算法和NURBS曲线的也是一样。

**4 细分建模技术设计实现**

本文使用上面介绍的NURBS曲线曲面的细分算法，并通过编程将其实现。

**4.1 整体架构设计**

本文通过建立一个小型NURBS建模系统的软件，在软件中实现了NURBS细分建模的功能。本文选择Visual Studio 2015作为应用程序开发工具，使用Qt作为图形用户界面应用程序开发框架。

此小型的NURBS建模系统主要实现NURBS细分建模的功能，附加包含实现三维物体线框模型动画技术、三维物体线框模型的消隐技术、三维物体光照模型阴影的绘制技术。

**4.2模块功能设计**

功能：加载图片文件，对图片中的物体进行描边；或者自定义物体描边。描边完成之后获得描边数据,然后使用其数据进行细分建模，建立出物体的三维模型。

使用的流程图如图4.1所示：

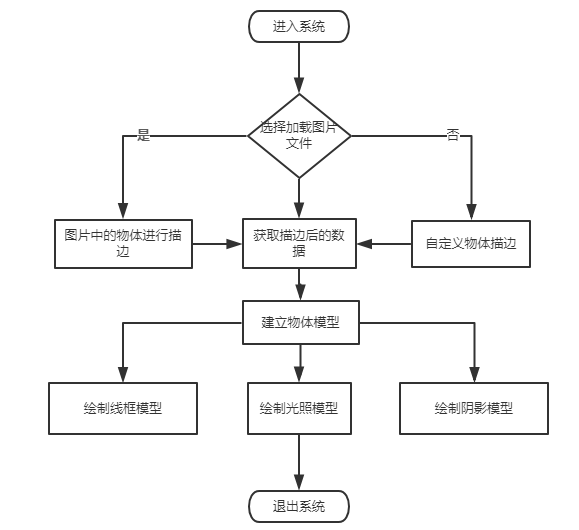


图4.1物体建模的流程图

**4.3 NURBS建模系统**

**4.3.1系统主要功能界面**

1. 主界面

图4.2 主界面

1. 加载图片建模的功能界面

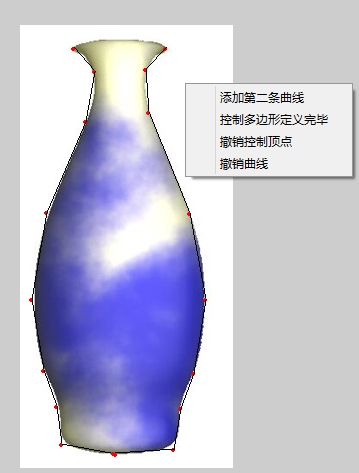
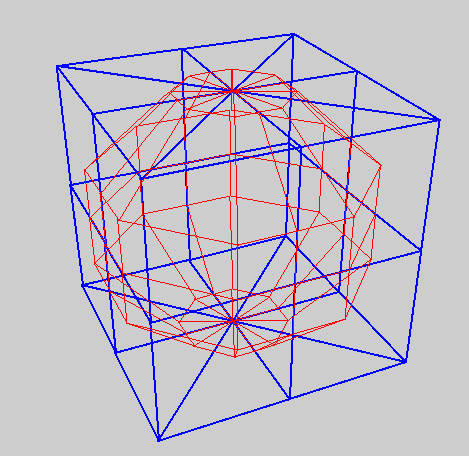
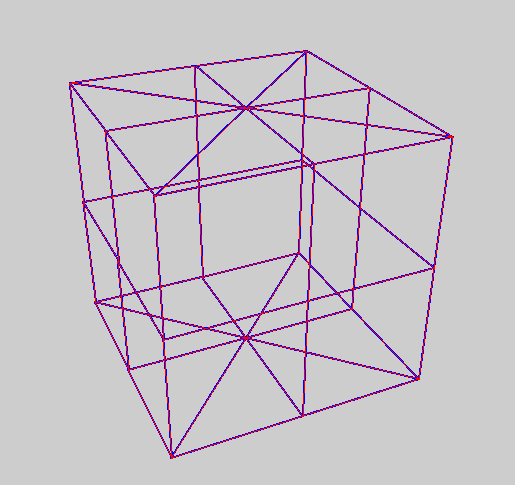


图4.3加载图片建模的界面

1. ****正方体细分实现的过程效果:

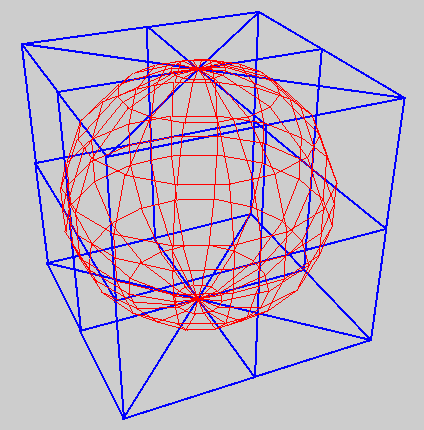
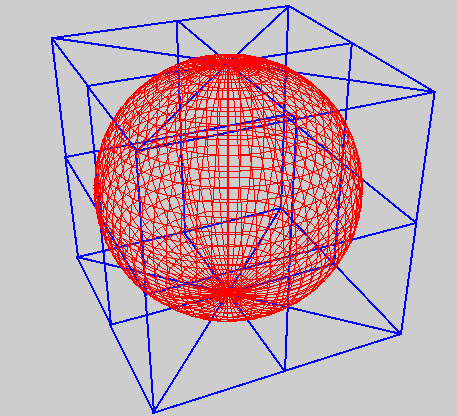


图4.5 细分一次效果

图4.4 细分零次效果

图4.7 细分四次效果

图4.6 细分二次效果

图4.5 细分二次效果

1. 通用物体细分实现的过程效果:

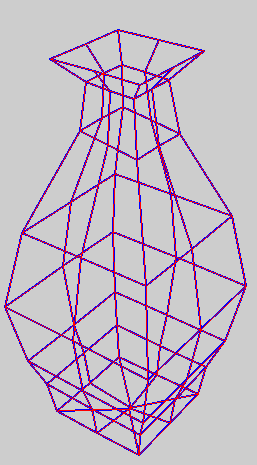
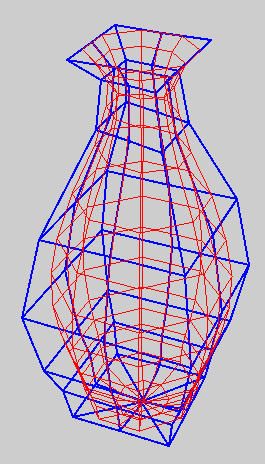


图4.9 细分一次效果

图4.8 细分零次效果

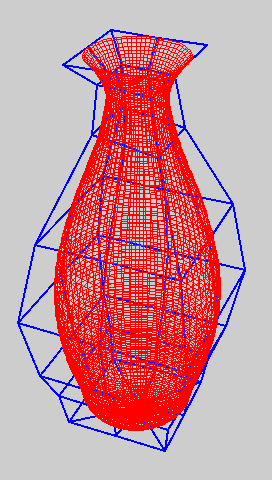
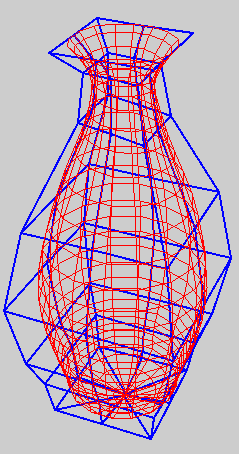


图4.11 细分四次效果

图4.10 细分二次效果

**4.4 基于Qt平台的设计**

Qt 是一个C++图形用户界面的应用程序开发框架，于1991年被Qt Company所开发，同时Qt是具有跨平台性的。Qt开发框架不仅能用来开发GUI的程序，还能用来开发非GUI的程序，像控制台工具和服务器就属于非GUI的程序。Qt是面向对象的框架，使用元对象编译器和宏生成扩展，使其真正实现组件式编程。

**4.4.1主要的函数介绍**

1. 节点矢量的细分函数

inTime 为细分次数，m\_Vertices为控制顶点，m\_HorizontalNodes为u向的节点矢量，m\_VorizontalNodes为v向的节点矢量，m\_W为权因子

void CGeneralSubdivision::SubdivideCurveKnots(unsigned int inTime)

{

//u向

typedef unsigned int uint;

for (uint uTimes = 0; uTime < inTime; ++ uTime)

{

vector<double> copyHNode;

uint countI = m\_Vertices.size();

for (uint indexI = 0; indexI < countI; ++indexI)

{

copyHNode.clear();

copyHNode.assign(m\_HorizontalNodes.begin(), m\_HorizontalNodes.end());

vector<double> tempNode;

tempNode.assign(copyHNode.begin(), copyHNode.end());

unsigned int countJ = m\_Vertices[indexI].size();

for (uint indexJ = c\_GeneralK; indexJ < countJ; ++indexJ)

{

// 二尺度细分

double node = (tempNode[indexJ + 1] - tempNode[indexJ]) / 2.0;

copyHNode.push\_back(tempNode[indexJ] + node);

sort(copyHNode.begin(), copyHNode.end());

unsigned int position = SubdivideCurveKnot(tempNode[indexJ] + node, copyHNode);

CurveKnotInsertVertex(tempNode[indexJ] + node, position, m\_Vertices[indexI], copyHNode, m\_W[indexI]);

}

}

m\_HorizontalNodes.clear();

m\_HorizontalNodes.assign(copyHNode.begin(), copyHNode.end());

vector<double> copyVNode;

countI = m\_Vertices[0].size();

uint countJ = m\_Vertices.size();

for (uint indexI = 0; indexI < countI; ++indexI)

{

vector<CVertices> formVertices;

formVertices.resize(countJ);

for (uint i = 0; i < countJ; ++i)

formVertices[i] = m\_Vertices[i][indexI];

vector<double> formW;

formW.resize(countJ);

for (uint i = 0; i < countJ; ++i)

formW[i] = m\_W[i][indexI];

copyVNode.clear();

copyVNode.assign(m\_VerticalNodes.begin(), m\_VerticalNodes.end());

vector<double> tempNode;

tempNode.assign(copyVNode.begin(), copyVNode.end());

for (uint indexJ = c\_ShpereK; indexJ < countJ; ++indexJ)

{

double node = (tempNode[indexJ + 1] - tempNode[indexJ]) / 2.0;

copyVNode.push\_back(tempNode[indexJ] + node);

sort(copyVNode.begin(), copyVNode.end());

unsigned int position = SubdivideCurveKnot(tempNode[indexJ] + node, copyVNode);

CurveKnotInsertVertex(tempNode[indexJ] + node, position, formVertices, copyVNode, formW);

}

if (indexI == 0)

{

m\_Vertices.resize(formVertices.size());

m\_W.resize(formVertices.size());

int count = m\_Vertices[0].size();

for (int i = countJ; i < formVertices.size(); ++i)

{

m\_Vertices[i].resize(count);

m\_W[i].resize(count);

}

}

for (uint i = 0; i < formVertices.size(); ++i)

{

m\_Vertices[i][indexI] = formVertices[i];

m\_W[i][indexI] = formW[i];

}

}

m\_VerticalNodes.clear();

m\_VerticalNodes.assign(copyVNode.begin(), copyVNode.end());

}

}

1. 节点矢量的细分后新顶点的计算函数

void CSubdivisionShpere::CurveKnotInsertVertex(

double inInsertKnot,

int inIndex,

vector<CVertices>& inOutVertices,

vector<double> inNodes,

vector<double>& inOutW)

{

typedef unsigned int uint;

vector<CVertices> tempVertices;

vector<double> tempW;

tempVertices.assign(inOutVertices.begin(), inOutVertices.end());

tempW.assign(inOutW.begin(), inOutW.end());

inOutVertices.resize(tempVertices.size() + 1);

inOutW.resize(tempW.size() + 1);

// 处理插入节点后的控制顶点

for (uint j = inIndex - c\_GeneralK + 1; j <= inIndex; ++j)

{

double numerator = inInsertKnot - inNodes[j];

double denominator = inNodes[j + c\_GeneralK + 1] - inNodes[j];

double aj = 0.0;

if (0 == numerator || 0 == denominator)

aj = 0.0;

aj = numerator / denominator;

inOutVertices[j] = aj \* tempW[j] \* tempVertices[j] + (1 - aj) \* tempW[j - 1] \* tempVertices[j - 1];

inOutW[j] = aj \* tempW[j] + (1 - aj) \* tempW[j - 1];

inOutVertices[j] = inOutVertices[j] / inOutW[j];

}

for (uint j = inIndex + 1; j < inOutVertices.size(); ++j)

{

inOutVertices[j] = tempVertices[j - 1];

inOutW[j] = tempW[j - 1];

}

}

1. Qt绘图函数

void CTestView::paintEvent(QPaintEvent \* event)

{

QPainter painter

//绘制功能只要使用painter进行。

}

**4.5 主要的类说明**

1. 系统主界面类：Test

该类主要实现界面设计，有菜单栏、工具栏、状态栏、右键菜单、快捷键等功能。

1. 绘制三维视图的类：TestView

该类主要实现绘制二维、三维物体、鼠标消息以及键盘消息等功能。

1. 三维透视投影类：Projection

该类主要实现三维物体坐标转换为二维坐标的屏幕显示。

1. 获取NURBS控制顶点类：CurveControlVertices

该类主要实现获取NURBS的控制顶点。

1. 生成细分曲面类：GeneralSubdivision

该类主要实现使用CurveControlVertices类获取的控制点数据用于生成细分曲面需要的数据，进而用于进行绘制细分曲面。

1. 细分正方体类：SubdivisionSphere

该类主要实现将正方体细分为一个球的功能。

**5 结论**

**5.1已完成的工作**

本文通过对NURBS建模技术的研究，实现了使用NURBS技术进行三维建模，同时通过代码建立了一个小型的NURBS建模系统。在完成NURBS曲面细分技术的研究之后，将其研究成果加入NURBS建模系统中，实现了NURBS节点插入细分算法，实现了NURBS曲面细分建模技术。

**5.2今后进行的工作**

NURBS曲面采用的节点插入来进行曲面细分，但是薛翔指出新的控制点总是整行、整列的加入到曲面上，因此细分后的NURBS曲面不够精细[21]。为了解决NURBS曲面细分后不够精细的问题，人们提出了T样条。T样条曲面是由Sederberg等在2003年7月的Siggraph会议上提出[22]。Sederberg定义的T样条曲面和之前一直使用的张量积曲面定义不同，其使用的是一种新的点样条曲面(Point-based Splines)定义方式[22]，因此，曲面的结构变得更加灵活，使样条曲面真正的具有了局部细分的特性。Sederberg在2003年的文章中分别介绍了T网格的结构组成、T样条曲面定义、局部均匀细分算法、并改进了B样条曲面合并法[22]。在2004年他在文章中又给出了更灵活的局部非均匀细分算法，并将该算法用于NURBS曲面模型的精简处理[23]。但同时也是因为T样条曲面局部细分算法过于灵活，细分后的T样条曲面并不总是具有仿射不变，这是曲面建模人员不想见到。在后续的研究工作中，将对T样条曲面建模进行研究，并研究如何解决NURBS细分和T样条曲面细分中不总是具有仿射不变的问题。

参考文献

[1] Barnhill R. E.,Riesenfeld R. F.. Computer Aided Geometric Design[M]. Academic Press

,1974.

[2] Chaikin G. M.. An algorithm for high-speed curve generation[J]. Computer graphics and image processing，1974，3(4): 346-349.

[3] Catmull E.,Clark J... Recursively generated B-spline surfaces on arbitrary topological

meshes[J]. Computer-aided design,1978,10(6): 350-355.

[4] Doo D.,Sabin M..Behaviour of recursive division surfaces near extraordinary points[J].Computer- Aided Design,1978,10(6): 356-360.

[5] Loop C.Smooth subdivion surface based on triangles[D].Salr Lake City:University of Utah,1987.

[6] DynN.,Levine D.,Gregory J. A.. A butterfly subdivision scheme for surface with tension control[J]. ACM transactions on Graphics(TOG),1990,9(2):160-169.

[7] Sederberg T. W.,Zheng J.,Seewll D.,et al. Non-uniform recursive subdivision surface[C].ACM,1998.

[8] Sederberg T. W.,Zheng J.. Bakenov A.,et al. T-splines and T-NURCCs[J]. ACM transactions on graphics(TOG), 2003, 22(3):477-484.

[9] Stam J.. Exact evaluation of CatmullClark subdivision surfaces at arbitrary parameter values[C]. techniques,ACM,1998: 395-404.

[10] Schoenberg L. On spline funetion[A]. //Shisha o. ed. Inequalities[C]. NewYork: Academic Press,1967: 255 - 291.

[11] De Boor C. On Caleulation with B-splines[J],J. appox. Theory,1972.

[12] Gordon W J,RiesenfeldR F.B - spline eurves and surfaces[A].// Computer - aided geometrie design[C]. Salt Lake CÏty:Academic Press,1974.

[13] 施法中.计算机辅助几何设计与非均匀有理B样条[M].北京:高等教育出版社,2013．

[14] Charles Loop. Smooth Subdivision Surfaces Based on Triangles[M]. Mathematics thesis. University of Utah,1987.

[15] Dyn N.,Levin D.,GregoryJ A. A Buytterfy Subdivion Scheme for SurfaceInterpolation with Transion Control[J]. ACM Trans. Gr,1990,9(2).

[16] 于行洲,查红彬,石青云.蝶形细分面片的光顺[J].中国图象图形学报,2003,8(3):299-305.

[17] ZorinD.,Sho derP.. Subdivision for Modeling and Animation[C] . SIG2 GRAPH 2000 Conference Proceedings.Course Notes.20001472115.

[18] D. Doo: A subdivision algorithm for smoothing down irregularly shaped polyhedrons, Proceedings on Interactive Techniques in Computer Aided Design,1978,pp. 157 – 165.

[19] 彭小新,唐月红.自适应T样条曲面重建[J].中国图象图形学报,2010,15(12):1818-1825.

[20].B.K. Choi,C.S.Lee et a1. Compound Surface Modeling and Machining. Computer Aided.

[21] 薛翔. T样条曲面造型技术的研究[D]. 南京航空航天大学,2014.

[22] Thomas W. Sederberg, Jianmin Zheng,Almaz Bakenov,et al. T-Splines和T-NURCCS. ACM Trans, 2003.

[23] Thomas W. Sederberg, Jianmin Zheng,Tom Lyche,et al. T-Splines简化和局部重组. ACM Trans,2004.

**致谢**

不知不觉之间，毕业设计的工作即将结束，在这里首先我要感谢我的导师孔令德教授，孔令德教授在我的课题选择中给予了很多的建议和指导，还为我制定了详细的毕业设计任务书，在毕业设计过程中还不断为我所遇到的问题进行解惑，这极大的促进我毕业设计的完成。孔令德教授对工作的认真负责、对学术的钻研精神和严谨的学风，都是值得我终生学习的。

在这四年的学习期间，遇到很多老师，很感谢老师们的悉心教导，让我学到了专业的计算机知识，掌握了扎实的专业技能。同时在四年期间，还遇到了很多一起学习图形学的伙伴们，然后一起加入我们系的研究所学习图形学，也为这次毕业设计作了重要的知识铺垫。

感谢研究所提供的学习环境，在研究所的两年学习时光让我受益终生。即将毕业，在以后的学习和工作中，定不会辜负老师与学校的教诲，实现自己的目标，做一个有价值的人。