

摘 要

虚拟现实是在计算机中营造的一个虚拟环境。它通过实时的、立体的三维图形显示、声音模拟、数据手套等各种传感设备、自然的人机交互界面使用户产生身临其境的真实感觉，是一个基于计算信息的沉浸式交互系统。它使观众参与到场景中来，而不仅仅是进行远距离观察。虚拟现实技术实时显示的要求对计算机软硬件技术和计算机图形学提出了挑战。一些高速的图形硬件以及有效的提高绘制速度的图形学算法应运而生。为解决计算机硬件有限的绘制能力与场景模型的无限复杂性之间的矛盾，计算机图形学的研究人员在相关方面进行了大量的研究。

多分辨率模型技术是一种对场景中所含景物细节的多层次表示方法。它通过建立原始场景模型的多个近似简化模型，表示原始模型不同程度的细节。利用该技术，绘制系统可以在保持视觉等价的前提下，选择最粗的层次模型，提高绘制速度。

本文对多分辨率模型技术进行了深入的研究，主要研究成果如下：

首先，为了能方便快捷地简化网格模型中冗余的共面面片，我们提出了基于高斯球的近似共面合并多分辨率模型。通过对高斯球面的层次分割对表面法向近乎一致的近似共面面片进行合并和简化，生成一系列不同细节层次的简化模型。该算法不仅可以有效保持模型几何特征，而且算法简单，适用广泛，易于实现。

其次，考虑到目前在很多应用领域广泛使用的网格模型实际上是曲面模型离散化、线性化的结果，本文提出通过对网格模型对应曲面的重新采样生成简化模型的思想。此外，对于采用启发式简化误差测量的增量式简化算法，由于在模型简化的不同阶段简化模型表现出不同的特点，对简化操作有不同的要求。我们提出采用可调加权控制函数，在简化过程中通过改变权值体现不同的简化要求，为

模型简化算法提供了新的思路。本文所提出的基于分割插值的多分辨率模型以及基于隐函数插值的多分辨率模型都是从以上思想出发实现了对模型的简化。✓

其中，基于分割插值的多分辨率模型利用插值分割技术通过对边相邻顶点的插值使边折叠目标点近似为曲面模型的采样点，大大减小了简化模型的 Hausdorff 误差。基于隐函数插值的多分辨率模型则通过对原始网格模型进行分区域变分插值，得到网格模型对应的变分隐函数曲面模型。在模型简化过程中，通过对该隐函数曲面的采样得到边折叠目标点。由于区域插值隐函数曲面在模型简化过程中保持不变，消除了由于模型简化而导致的对应插值隐函数曲面的累积误差。

关键词：虚拟现实，多分辨率模型，多层次细节，网格模型简化，高斯球，分割插值，变分隐函数曲面

Abstract

Virtual Reality is a Virtual Environment created by computer. It refers to an interactive system based on computer information, which brings users illusion of immersive experiences via real-time stereoscope display, sound system, data glove and other haptic devices. It permits the viewer participates in the environment, but not perceive it in distance. Real-time display, which is one of the bases of Virtual Reality technology, challenges the computer technology of software and hardware, and also the computer graphics. High-end computer system and effective algorithms for improving the performance of rendering system is worked out to meet the need of it. Though the current Current high-end graphics systems are capable of rendering tens of millions of polygons per second and support advanced rendering features in real time, the complexity of large geometric datasets appears to be growing at a faster rate as compared to the rendering capabilities of the graphics systems. Many software techniques are preferred to handle such large geometric datasets and render them at interactive rates on current graphics systems.

Multiresolution modeling is a multi-level representation method for a environment model. A series of approximation models are created which represent different level of detail. Then, the rendering system can choose the coarsest model that is appearance preserving to improve the performance.

This thesis makes a deeply research on the technology of multiresolution modeling, and has achieved some new effective results as the following:

A Gauss sphere based near coplanar merging multiresolution modeling algorithm is presented which can remove the redundant near coplanar facets from the triangle mesh conveniently and quickly. On the basis of a hierarchically partitioned Gauss Sphere, it generates the basic simplified model. Furthermore, a series of approximation model containing different level of details are created by further processing the basic simplified model according to the Gauss Sphere. This algorithm can not only keep the

geometric features of the model effectively just as other near coplanar merging based simplification method, but also be convenient to implement and has more wide application.

On the other hand, thinking of the mesh model used widely in many applications is in fact the discrete and linear approximation of the surface model, in this thesis we present the idea that generate the simplified model by resampling the corresponding surface model. Moreover, for incremental simplification algorithm that measures the contraction error heuristically, an adaptive measure scheme is need because the simplified model has difference characters in different simplification degree. An adaptive weighted control function is presented in this thesis, which present a new approach for model simplification algorithm. The subdivision interpolation based multiresolution modeling algorithm and the implicit function interpolation based multiresolution modeling algorithm simplify the mesh model on the basis of the above ideas.

The subdivision interpolation based multiresolution algorithm use the interpolation subdivision technology to compute the contraction target point by interpolating the vertices in neighbourhood, which leads to much smaller Hasdorff error. The implicit function interpolation based algorithm generates the regional variational implicit surface of the mesh model by implementing the variational interpolation technology on partitioned mesh model. And the edge contraction target point is got by resampling the corresponding region implicit surface. For the regional implicit surface keeps the same in the duration of simplification, the accumulated error is avoided.

Key Words: Virtual Reality, Multiresolution modeling, multi-level of details, Mesh Simplification, Gauss Sphere, Interpolation Subdivision, Variational Implicit Surface

第一章 概述

千百年来，人类一直在探索征服自然、开拓视野的道路。科学技术正是开启这扇神秘之门的钥匙。刚刚过去的二十世纪，是科技发展的黄金时代。不断涌现的新技术不仅加速了生产力的发展，给人们的生活带来了巨大的变化，更使科学技术本身以前所未有的速度发展起来。特别是计算机的出现，开辟了科技发展的新纪元。

1.1 虚拟现实技术

五十年前，计算机还仅能用于计算数学公式。然而，天才的计算机研究人员就已经预见到，计算机将带给人们一个色彩斑斓的世界。电影艺术家们更是对未来有着无限的幻想。在全息电影和立体声电影轰动全球的五十年代中叶，Morton Heilig发明了一种称为*Sensorama*的三维电影。*Sensorama*不仅能给人以立体的视觉、听觉享受，还在嗅觉、触觉等方面进行模拟，把观众带入神奇的虚幻世界。虽然*Sensorama*并不是一个交互的系统，不能称为我们现在所认识的虚拟现实，但它预言了虚拟现实系统的很多特性[Krueger91]。

1965年，计算机图形学的创始人Ivan Sutherland 在IFIP会议报告中提出了计算机窗口的概念[Sutherland65]，从学者的角度揭示了计算机生成虚拟世界的可行性，并在1968年发明了第一个三维头盔显示器，奠定了三维立体显示技术的基础。之后，由于虚拟现实在军事方面的重大意义，美国军方展开了一系列的研究。但由于其费用昂贵，一直没有得到实际应用，也使研究的广泛性受到限制。一直到八十年代末九十年代初，随着计算机处理技术的突飞猛进和图形绘制技术、数字信号处理技术、传感技术、人机接口技术等全面提高，国际上才真正形成了虚拟现实的研究热潮。虚拟现实也从此成为信息科学中非常活跃、并极具潜力的前沿研究领域。

虚拟现实，体现出其虚拟和现实的二重性。一方面它是高技术的产物，从物

理意义上讲它是不存在的；另一方面在人的感官世界里，它又是真实存在的，可以通过视觉、听觉、触觉等感知器官来感知。



图 1.1 虚拟自行车

Sutherland曾在他的IFIP会议报告“The Ultimate Display”中给VR，也就是他所说的“终极显示器”作了这样的定义，“终极显示器当然应该是这样的一个空间，计算机可以控制这一空间中所有陈设。这里的椅子真的可以让人坐下，手铐可以限制人的行动，而射来的子弹也可以导致致命的伤害。”[Sutherland65]。三十多年后的今天，Sutherland的预言成为了现实。VR技术已经用于飞行模拟、风洞实验、军事演习、医学诊断、文化娱乐等许多实践领域。但目前，还没有一个公认的虚拟现实（Virtual Reality—VR）的定义。一般来说，虚拟现实是在计算机中营造的一个虚拟环境。它通过实时的、立体的三维图形显示、声音模拟、数据手套等各种传感设备、自然的人机交互界面使用户产生身临其境的真实感觉，是一个基于计算信息的沉浸式交互系统（图1.1）。它使观众参与到场景中来，而不仅仅是进行远距离观察。

VR将给人们的生活带来巨大的变化。我们可以进入虚拟环境中，与朋友进行远程约会，或与奇妙的虚拟生物戏耍，还可以带上头盔显示器和数据手套，让自己沉浸在一个在真实世界永远无法达到的境界。VR技术，给工程设计人员、科学研究者和用户提供了一个进入计算机世界的窗口。在虚拟世界中，他们具有完全控制虚拟环境的能力，可以看到现实世界中早已发生、正在发生或尚未发生的事件，甚至可以看到在真实世界中不可能看到的东西……人类征服自然、拓展视野的种种设想可以在虚拟环境中进行预演。

VR有着广泛的应用前景，可以应用于商业、教育、娱乐等很多方面。建筑师可以带领用户参观正在设计中的建筑物；外科医生可以在虚拟手术中排练手术方案；交互的游戏则可以带给游戏者前所未有的真实感觉。此外，VR也将成为工程师和科学工作者强大的研究工具，带来对科学和工程学领域中许多挑战性问题的深入认识和解决。虚拟现实技术可以应用于大型工程试验和新产品的试制。由于在虚拟环境中能对材料、功能、结构进行全方位的验证并以最直观的方式显示仿真结果，并能反复演示仿真过程，且其仿真环境很容易根据设计人员的需求予以重建，可以大大减少甚至取代实物试验。VR使人类可以跨越时间和空间，辅助人们探索从未亦无法涉足的未知世界。这不仅大大拓展了人类的生理活动空间，而且为人类的知识创新提供了重要工具。虚拟现实提供的多通道信息融合和高维信息可视化的演示媒体，将人们的视野由二维(屏幕)扩展到三维(空间)，使科技人员更容易理解和洞察空间数据分布的三维特征，并通过对这些采自不同时刻，以不同形式呈现的信息的综合、分析和加工，揭示这些数据中蕴含的科学规律。这在遥感遥测数据的分析和融合、环境保护、灾情预测，气象分析、地质勘探等方面有广阔的应用前景。

虚拟现实技术本身涉及了计算机图形学、计算机视觉、心理学、传感器、力学、声学等许多领域。它通过集成视觉显示、跟踪系统、输入设备、语音系统、触觉反馈系统以及图形学和计算机硬件等多项技术，使用户产生置身于虚拟环境中的感觉。

虚拟现实有三个赖以存在的关键因素即实时性(Real Time)、沉浸性(Immersion)、和交互性(Interactivity)。所谓实时性是指虚拟环境能按用户当前的

视点位置和视线方向,实时地改变呈现在用户眼前的虚拟环境画面,并在用户耳边和手上实时产生符合当前情景的听觉和触觉/力觉响应。所谓沉浸性是指用户所感知的虚拟环境是三维的、立体的,其感知的信息是多通道的。所谓交互性是指用户可采取现实生活中习以为常的方式来操纵虚拟场景中的物体,并改变其方位,属性或当前的运动状态。

虚拟现实技术应用了很多异于传统人机界面的新型设备:

- **立体视觉显示器(Stereo Display):** VR的显示设备通过分别向用户的左右眼展示计算机生成的模拟人的左右眼睛观察的场景,产生三维立体视觉(Stereoscope)。如现有的快门眼镜(Shutter glasses)以及头盔显示器(Head Mounted Display)等。
- **头部跟踪器(Head Tracker):** 为了实现虚拟环境的沉浸感,两眼的画面需要随眼睛的位置以及视线方向的变化而实时地改变。因此需要头部跟踪器来感知视域的变化。VR的头部跟踪器通常配备在头盔显示器上。VR还用眼睛鼠标等跟踪视域。
- **手部跟踪器(Hand Tracker):** VR系统运用了比鼠标或操纵杆更自然的系统控制设备——数据手套。数据手套通过跟踪手和手指的位置及朝向,对系统进行控制,如按虚拟按钮、抓取、移动虚拟物体等。最新的控制方法还有眼睛控制(眼标)和思维控制等。
- **力反馈设备(Force Feedback):** 系统通过力反馈设备、触觉反馈器(Haptic Feedback)使用户感知虚拟环境的力反馈信息。触觉反应用户不仅能看到还可以摸到虚拟景物。
- **CAVE (the Computer Animated Virtual Environment):** CAVE是一种基于投影的环屏显示设备,它看上去象一个墙壁和地板都是计算机显示器的房间。CAVE的投影图象通过不断在左右眼图象间切换,实现立体显示。

这些特殊的输入输出设备,使虚拟现实成为非常接近真实世界的人机交互系统。

虽然虚拟现实发展至今，已经取得了很多重大的研究成果，并在很多领域得到了较为成功的应用。但在目前的技术水平下的虚拟环境中，人们还不能象在真实世界中那样自然、方便的进行各种活动。即使如此，也不能影响虚拟现实发展和应用的广阔前景。有科学家和企业家预言，VR将引起未来人类生活的重大变革。

然而VR技术的普及，到目前为止，仍在虚拟场景建模与真实感知、人与虚拟环境的自然实时交互、分布式虚拟环境的通讯机制与时空一致性、系统延迟等方面存在若干尚未解决的问题。

眼睛是人类感知环境最敏感的器官，视觉是人类获取外界信息最主要的通道。很自然地，VR对三维视觉界面提出了很高的要求。虚拟场景必须以一定的刷新频率实现实时动态显示，画面显示的任何滞后都会严重破坏VR的沉浸感。然而，由于虚拟环境具有无限复杂性，用传统的计算机图形学方法根本不可能达到对虚拟环境实时绘制的要求。于是快速漫游技术的研究热潮应运而生，人们试图采用新的场景建模与绘制技术来解决这一传统计算机图形学无法解决的矛盾。本文就是从这一新的角度对三维虚拟环境的实时显示技术进行了探索。

1.2 虚拟环境的快速漫游

在虚拟现实系统中最耗费时间的一项任务是图像生成。除VR之外，其它很多应用领域也需要快速图形计算：如计算机辅助设计（Computer Aided Design）、科学计算可视化、娱乐、训练等。

因为20帧/秒(20fps)是人脑把一系列连续的静止图像合成平滑动画的下限，在VR系统中，系统至少需要维持20帧/秒的帧频才能保证虚拟场景画面的连续性。为满足绘制动态场景的要求，目前大多数VR系统的显示器的帧频为30Hz。为维持这样的帧频，系统需要花费很大的计算量。且计算量还会随场景的复杂程度的增长而变大。尽管传统图形学的很多绘制技术(如光线跟踪和光线辐射方法)可以产生具有阴影、高光、以及透明等效果的极具真实感的图像，但它们生成一帧高度复杂场景的图象可能需要几个小时，很难满足实时性需求。为此，大多数VR绘制过程都采用了专用硬件，来提高绘制速度。近年来，对图形生成硬件加速器的需求不断增长。目前有很多厂商在销售针对个人机的图像生成器，其中大

多基于 Intel i86 处理器。SGI 在生产图形工作站方面获得巨大的商业利益，在很多 VR 实验室和高端 VR 系统中得到了广泛的应用。高性能图形系统已经具有每秒上千万个面片的绘制能力[Molnar92]，能支持实时光照、纹理映射、反走样等高级绘制过程。

3D图形硬件技术的高速发展，似乎可以成为解决实时绘制问题的钥匙。但实际上，人们获得大规模复杂几何数据的必要性和能力也并不是处于静止状态。现代科技的发展使人们可以方便地构造越来越复杂的场景模型，大型数据模型可能包含上亿个面片（多边形或曲面片）。如波音777客机模型由5亿个面片组成，如果直接采用传统的绘制流程，需要具有每秒一百亿面片绘制能力的图形系统才能实现20帧/秒的帧频。对于这类复杂场景，即使在象SGI Infinite Reality Engine和PixelFlow这样代表当今图形硬件最高水平的图形工作站上，也不能实现交互显示。

近年来，人们为在现有的图形硬件条件下实现复杂场景的实时交互显示，在加速绘制、大规模数据管理、碰撞检测、系统集成等软件技术方面作了很多研究和探索。本论文将主要研究加速绘制技术，而不涉及大规模数据管理、碰撞检测、系统集成等方面的内容。

1.2.1 加速绘制技术

目前人们对加速绘制技术的研究主要集中于基于图象的绘制、模型简化、可见性剔除、高阶对象的直接绘制、纹理映射等几个方面[Manocha99]。这些方法通过动态删除用户在当前视点和视线方向下看不到的部分场景或在保持视觉等价的前提下删除场景中的某些难以分辨的细节来减少需要绘制的图形元素，提高绘制速度，实现实时显示的目的。

- 基于图象的方法[Shade96, Shade98]：通过纹理映射、3D 图象变形等技术，用图象表示复杂场景。这样，场景的绘制速度独立于场景的复杂程度，而只与屏幕的大小有关。
- 模型简化[Clark76, Hoppe96]：当物体远离视点时，对其中细小的、不被

注意的细节，以较低的分辨率绘制。

- 可见性剔除[Zhang97]: 快速剔除场景中的大部分不可见面，只绘制可能可见的面片。
- 高阶对象的交互绘制[Kumar99]: 对场景模型中的样条曲线、曲面等高阶对象进行实时动态分割绘制。

1.2.2 可见性剔除

在真实世界中，由于视域、朝向、遮挡等原因引起场景中一部分景物不可见是非常常见的。同样，在虚拟环境中，也有很大一部分面片相对于当前视点不可见。这就是说，对于给定帧画面，只需要绘制场景的一小部分。如果用传统的硬件深度缓冲器图形系统进行消隐计算，尽管系统最终也可以正确地判断景物可见性，但由于直到绘制流程的最后才确定其可见性，大量前序计算不可避免，从而导致系统资源的极大浪费。可见性剔除算法就是把不可见的场景从其它场景中区分出来，使进入硬件绘制的面片尽可能减少，达到快速绘制的目的。

针对视域外面、背向面、被遮挡面三种不可见面，可见性剔除算法相应分为视域裁剪 (View-frustum Culling)、背向面剔除 (Back-face Culling)、遮挡剔除 (Occlusion Culling) 三类。

目前，大多数漫游系统通过进行动态视域裁剪 (View Frustum Culling)，快速剔除当前视域四棱锥之外的场景，使需绘制的图形元素得到了一定程度的减少。Airey等人提出了针对结构性强的虚拟环境（如建筑物）的可见性算法。他们把整个模型空间用墙壁、门等划分成很多单元[Airey90, Teller91, Luebke95]，并基于划分进行可见性计算。由于建筑物本身由一系列房间组成，这类方法非常适用于建筑模型的漫游。但是这种单元分割法具有一定的局限性，对城市模型等任意结构的模型不能进行快速有效的可见性剔除处理。Greene等则提出了分层Z-buffer算法[Greene93]。算法建立物体空间八叉树的同时，在图象空间建立Z-pyramid层次结构，利用物体空间、图象空间和时间三个方面的连贯性，加速可见性剔除计算。其算法复杂度对模型复杂程度的依赖性大大降低。但是，商业图形系统并

不具有支持算法中的Z-pyramid的能力，而单纯软件实现该算法开销很大，从而损害了该算法的实用性。后来，Coorg和Teller[Coorg96]以及Hudson[Hudson97]等又提出了在景物空间的多边形模型的空间可见性算法。但这种算法仅限于遮挡物全为凸几何体的情形，而且不能对模型进行大幅度的剔除。1997年，Hansong Zhang等提出了分层遮挡图算法[Zhang97]，在绘制每一帧图象时动态选择遮挡物，建立分层遮挡图（Occlusion Maps）对遮挡效果进行累计和合并，用于剔除相对于当前视点不可见的几何体，减轻绘制系统的开销。

1.2.3 高阶曲面的交互显示

实际应用中，很多模型是由 NURBS 曲面、旋转曲面、放样曲面等样条曲面表示的整体曲面模型。大多数绘制系统在绘制这些曲面模型时，通常先将它们离散成平面片然后进行平面绘制。这种方法由于采样密度固定，在实时绘制时常会出现多边形模型过粗（视点较近时）或过细（视点较远时）的情况。此外，人们还采用实时离散的方法绘制曲面，但以传统的离散算法，在交互帧频下只能绘制包含 1000 个样条曲面的模型[Abi-Ezzi94, Abi-Ezzi91, Luken93, Rockwood89]。近年来，Kumar 等人通过利用相邻帧画面的连续性、可见性剔除、面片的增量式分割、并行处理等技术[Kumar97, Kumar95b, Kumar96]，大大改善了交互帧频下高阶曲面的绘制能力。目前，已经能够交互绘制包含近 4000 个样条曲面的模型。

1.2.4 基于图象的方法

图形学中，用图象代替几何的思想由来已久。早在1974年Catmull就提出了纹理映射（Texture Mapping）的方法[Catmull74]，开创了基于图象的图形绘制的先河。之后，相关的研究如火如荼，Blinn和Newell将周围的环境事先绘制，并将其映射到有光泽、具有反射性的物体表面上[Blinn76]，这就是人们所说的环境映照（Environment Map）。环境映照最初是作为光线跟踪的一种廉价替代。它基于表面光泽的物体会反射出周围环境的现象，将周围环境图象预先计算好存储起来，当绘制该物体时，直接用纹理映射的方法来产生环境的映射效果，从而避免了光线跟踪的复杂过程。SGI Performer Town则用纹理表示远距离物体的表面细节，以增强视觉效果。

Regan 和 Pose [Regan94]创造了一种硬件系统，采用纹理代替背景，而对前景物体进行优先绘制，并用图象合成的方法生成新一帧画面，以保证较高的画面刷新频率。Torborg 和Kajiya [Torborg96] 则提出Telisman体系结构。在这种结构中，可以进行几何以及图象的绘制，其中，图象是通过仿射变换进行变形处理。Telisman把场景划分区域，一些移动物体的绘制结果（图象）可以通过Warping应用于多帧画面。但它并不能自动选择哪些场景用图象表示。

以上的方法在场景的绘制过程中，用纹理表示视觉重要度低的物体，可以在一定程度上提高绘制速度。但由于它们依赖于分离场景和对物体的交互选取，不适用于一般应用。

近几年来，人们开始探索直接用图象作为输入进行场景表示和绘制的方法。

基于图象的建模和绘制技术（Image Base Modeling and Rendering—IBMR）就是这样应运而生的。IBMR用图象代替传统的几何输入来进行建模和图象合成。它用预先获得的一组图象（合成的或真实的）来表示场景，而通过适当地组和原有的图象来合成新图象。基于IBMR技术的场景绘制速度彻底摆脱了对场景复杂度的依赖，而只与显示分辨率有关。

由于IBMR在绘制速度和画面真实感上的优势，吸引了大批图形学界人士的关注，获得了很多有意义的研究成果，现有的方法按表示模式与视图合成方法的不同可归纳为以下几类[Xu99]：

- 基于立体视觉的方法[Laveau94, McMillan95, Fuch94, Kanada95, Kang96, Koch95, Debevec96]：主要利用立体视觉（Stereo Vision）技术由已知的参考图象（已知的输入图象的有限分辨率集合——reference image）合成相对于新视点的理想图象
- 基于视域插值的方法[Chen93, Seitz95, Seitz96a, Seitz96b, Watt98, Werner95]（View Interpolation）：由参考图象线性插值生成一幅新的视图。新视图的视点位于两参考图象视点的连线（称为基线，Baseline）上。
- 基于图象拼合和分层的方法[Kang96, Szeliski94a, Szeliski94b, Szeliski96,

McMillan95, Kumar95a]: 将同一场景的多张互有重叠的图象组合成一幅较大的图象的处理称为拼合 (Mosaic)

- 基于全视函数的方法[Adelson91, McMillan95, Levoy96, Gortler96, Wong97, Li98a, Shun99]: Adelson用全视函数来描述空间中任意点在任意时刻、沿任意方向所看到的场景画面, 并将其定义在包含视点的空间位置 (V_x, V_y, V_z) 、视域方向的仰角 ϕ 和方位角 θ 、光波波长 λ 和时间 t 等的七维参数空间上:

$$\mu = \text{Plenoptic}(\theta, \phi, \lambda, V_x, V_y, V_z, t)$$

基于全视函数的方法, 试图捕获空间任意区域内的完全光流以重建该函数。

虽然以上方法可以在一定情况下实现加快绘制速度的目的, 但对于可见且视觉重要度较高的物体, 还是不可避免地需要进行基于几何的绘制。因此, 解决复杂模型的快速绘制问题十分必要。

1.2.5 模型简化

由于平面多边形, 特别是三角形的绘制简单、高效, 并且极易用硬件实现, 由三角形组成的网格模型是目前最为常用的一种模型表示形式。除虚拟环境外, 它还应用于有限元方法、电影特技等很多领域。人们可以通过激光深度扫描仪、计算机视觉系统、医学图象设备和计算机辅助设计系统等先进的现代科技手段获得复杂物体的多边形网格模型。这些网格模型通常通过密集采样表示光滑曲面或大规模场景, 因而非常复杂, 包含极其丰富的表面细节。其中, 由曲面重建和等值面抽取所获得的模型则通常包含均匀分布的非常稠密的曲面采样。

在对网格模型进行绘制处理时, 由于运行时间的限制, 通常需要控制模型绘制的精确度, 实现模型精确度和处理时间之间的折衷。此外, 由于图象分辨率的限制, 场景中可分辨可见多边形个数实际上也是有上限的。因而, 对在视平面上的投影很小的景物显然无需采用复杂的几何来描述, 只需采用简单的几何模型即可。

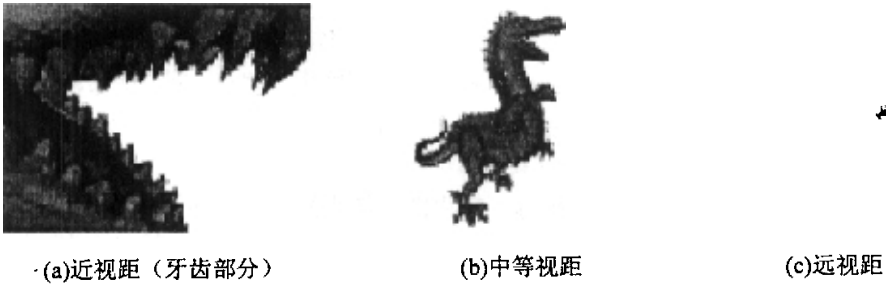


图 1.2 三个不同视距所绘制恐龙模型

如图 1.2b 所示，为由 100,000 个面片组成的恐龙模型。若对其进行近距离观察（图 1.2a），位于视域内的仅为整个模型的一小部分，可以观察到模型局部区域（牙齿部分）的精确细节。反之，如果进行远距离观察（图 1.2c），整个模型在屏幕上的投影只占很少的几个像素。显然，远距离情况下绘制的恐龙模型，已不能辨别任何细节，并不需要用 100,000 个面片来表示。

而模型简化最常见的应用就是降低稠密采样模型的复杂度。多分辨率模型的构造则使绘制系统可以在不同情况下针对待绘制景物的重要度以及对绘制时间的实时性要求，选用或重建适当的层次细节模型。

由模型简化建立的多层次细节模型最初由手工建立并应用于飞行模拟。在传统的飞行模拟器中，人们用手工的方法建立地貌的 LOD（Level of Detail）模型。LOD 模型可以在大规模地形的绘制中有效地控制需绘制对象的几何复杂度、提高绘制速度，但是手工建立大规模复杂场景的 LOD 模型，显然是一个无法想象的浩大工程，大大阻碍了这一技术的推广。近年来，人们开始探索利用多边形简化算法自动生成 LOD 模型的方法，很多有效模型简化算法相继被提出。一般来说，理想的模型简化算法应该具有以下性质：

- 保真性：简化算法所生成的模型，应与原始模型保持视觉等价。模型中的重要特征，即使在简化程度很大的模型上也应得到保持。
- 广适性：应适用于包括非流形曲面等的所有输入模型。
- 有效性：应能进行快速简化，而不需花费大量预处理时间。
- 大幅度简化：应该能进行包括拓扑简化在内的简化运算，实现对原始模型

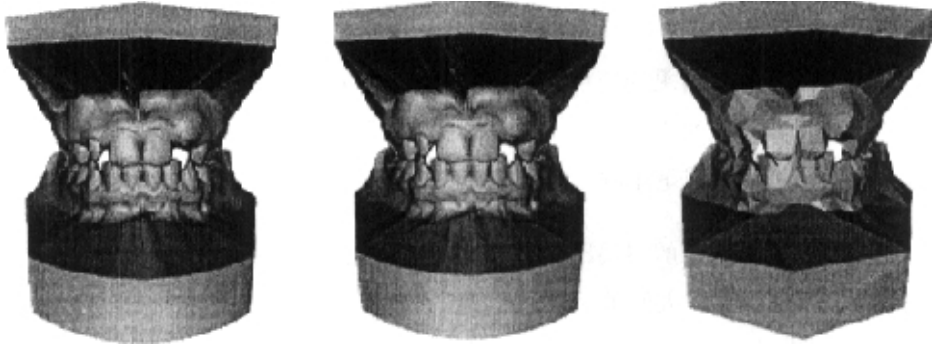
的大幅度简化。

- ◆ 对其他表面属性的考虑：除了模型的几何性质，优良的简化算法在简化过程中，应该控制模型的表面法向、颜色、纹理坐标等方面的误差。

本文将在1.3节，对近年来国际、国内图形学研究小组在通过模型简化算法自动生成多分辨率模型方面所取得的研究进展进行介绍。

1.3 多分辨率模型技术

多分辨率模型，顾名思义是一种对场景中景物的多层次表示方法。它通过建立原始场景模型的多个近似简化模型，表示原始模型不同程度的细节。细节越丰富，模型越复杂。



(a) 424,376 个面片(原始模型) (b) 60,000 个面片 (c) 1000 个面片

图 1.3 牙齿的多分辨率模型

如图 1.3 所示为牙齿模型及其两个简化模型。其中，图 1.3 (a) 所示为由四十多万面片组成的原始模型，直接由扫描仪扫描获得，采样密集，包含几乎所有的细节；图 1.3(b)所示模型只有原始模型面片数的 86%，所含细节较少，但在图中我们几乎分辨不出它与原始模型的差别；图 1.3 (c) 则只有 1000 个面片，模型的很多细节（如牙龈上的一些曲率变化特征）都已被简化，但仍保持了模型的总体结构特征。多种细节层次的模型的总体则构成了牙齿的多分辨率模型。在

需要提高绘制效率的应用中，可以用简化模型来代替原始模型，节省绘制时间。

多分辨率模型，虽然只在近几年来才开始得到人们越来越多的关注，但实际上它并不是一个很新的概念。早在二十多年前，Clark 就提出了多分辨率模型 (Multiresolution Modeling) 的思想[Clark76]。Clark 通过构造景物的多分辨率模型来表示复杂场景的不同层次细节 (Level of Detail)。而在对建立了多分辨率模型的场景进行绘制的过程中，绘制系统在保持视觉等价的前提下选择最粗的层次模型，从而能减小场景的复杂度，提高绘制速度。多分辨率模型表示技术在大规模地形数据模拟中的应用也在同一时期开始，并逐步发展起来。而直到进入九十年代以来，随着遥感技术的发展以及图形学在大型科学计算中的应用，人们越来越关注于一般多边形网格曲面场景的多分辨率模型的研究，在这一领域做了大量的研究工作。

多分辨率模型算法一个很重要的特征是否进行拓扑简化。在很多应用领域如医学图象相关应用中不允许拓扑的改变，对拓扑结构的保持甚至比对模型整体形状的保持还要重要。然而，也有一些领域刚好相反，只关心简化模型的视觉效果。因此，虽然进行拓扑简化可以使模型简化的程度增强，但并不能成为一种可以广泛应用的简化方式。本论文所提出的简化方法目前只限于进行保持拓扑的简化，但实际上它们可以很方便的拓展到支持拓扑简化的多分辨率模型。本文不对算法是否保持拓扑进行详细的讨论，而主要针对近年来所提出算法的简化策略以及简化过程的误差控制、简化结果的误差度量等方面进行讨论。

一般来说，多分辨率建模可以分为细化(Refinement)和简化(Decimation)两类方法。细化方法通过递归地向原始模型的最粗近似模型 M^0 中插入一点或多点，不断添加细节，获得不同分辨率的近似模型[Turk92, Hoppe96, Lounsbery94, Liu98]。而简化方法则刚好相反，它通过删除模型中的几何元素，去掉重要度较低的模型细节，生成低分辨率的简化模型。虽然两种方法细节变化的顺序不同，但二者有一个共同之处——均通过对原始模型的变换操作建立初始模型的简化近似模型。

1.3.1 细化的方法(Refinement Method)

细化方法是一种多步迭代算法。首先建立原始模型的基本近似模型 (Base Approximation) M^0 , 然后在 M^0 中插入一点或多点, 递归地执行下去, 提高初始建立模型的精确度, 直到获得所要求的误差或模型顶点数目, 生成较高分辨率的模型。实际上, 由细化方法建立多分辨率模型是一个从高度简化原始模型开始, 逐渐恢复其细节的过程。

Faugeras 在[Faugeras84]中提出通过不断向网格模型中插入新的顶点建立三维物体的近似多边形网格模型。算法以两个贴合的三角形作为3D物体的初始网格模型。网格模型中各三角面片分别对应一输入的顶点集合。算法查找各点集中与相应三角面片距离最远的点, 并判断该距离是否超过一设定的阈值。若超出阈值, 则在该点处插入新顶点, 并把三角形分为3到6个子三角形, 同时把该三角形对应的顶点集分为相应数量的子集, 与各子三角面片对应。递归插入新顶点, 则可得到不同细节层次模型。该算法是提出较早的一种细化算法, 其生成的简化模型的质量较差。

1994年, Delingette提出把误差、边长、曲率三项之和作为能量函数, 根据能量最小原理建立三维数据点的拟合曲面[Delingette94]。与Faugeras的算法类似, Delingette也是以一个与原始模型无关的简单的网格模型作为细化运算的初始模型。算法递归地对网格进行调整, 以使全局能量最小[Delingette92]。然后, 对调整所得的网格模型进行细化处理: 对曲率大、拟合误差较大或存在狭长的三角形的区域进行分割, 并把顶点移向高曲率的点[Delingette94]。该算法与[Hoppe93]的方法相比具有较高的执行速度, 但简化效果比较差。

此外, 基于小波分解 (Wavelet Decomposition) 的多分辨率模型[Lounsbery94][Eck95]和渐进网格模型[Hoppe96, Liu98]都属于细化的算法。

● 基于小波分解的算法 (Wavelet Decomposition)

多分辨率分析 (Multiresolution Analysis—MRA) 是小波最基本的特征之一, 它为建立物体的多分辨率表示技术提供了相当简洁的数学模型[Eric96]。通过小波

分析可以把曲面分解为拟合域和细节域。其中，拟合域为简化的模型基本形状；细节域则包括一系列曲面细节。结合拟合域和细节域可以生成一系列连续分辨率模型。小波分解已经成功地应用于生成信号及图象的多分辨率表示[Mallat89, Eric96]。Lounsbery则把多分辨率分析的概念推广到任意拓扑的模型[Lounsbery94, Lounsbery97]，提出对具有分割连通性的曲面进行小波分解。把小波这一有力的数学工具引入几何场景的多分辨率模型表示中，无疑是图形学研究的一项创举。但是，由于对曲面连通性的限制使算法在简化过程中必须维持曲面的分割连通性，限制了简化过程的优化，导致模型简化结果并不理想。此外，MRA不能有效地处理除基模型中特征边之外的棱边（Creases）[Hoppe96]。1995年，Eck等人提出了把任意曲面转换成分割连通的模型的方法，从而使MRA能够应用于任意连通的模型[Eck95]。但这种转换预处理过程显然会给基模型带来误差。另外，MRA也不能对原始模型的拓扑结构进行任何简化或修改变换。

● 渐进网格模型（Progressive Mesh）

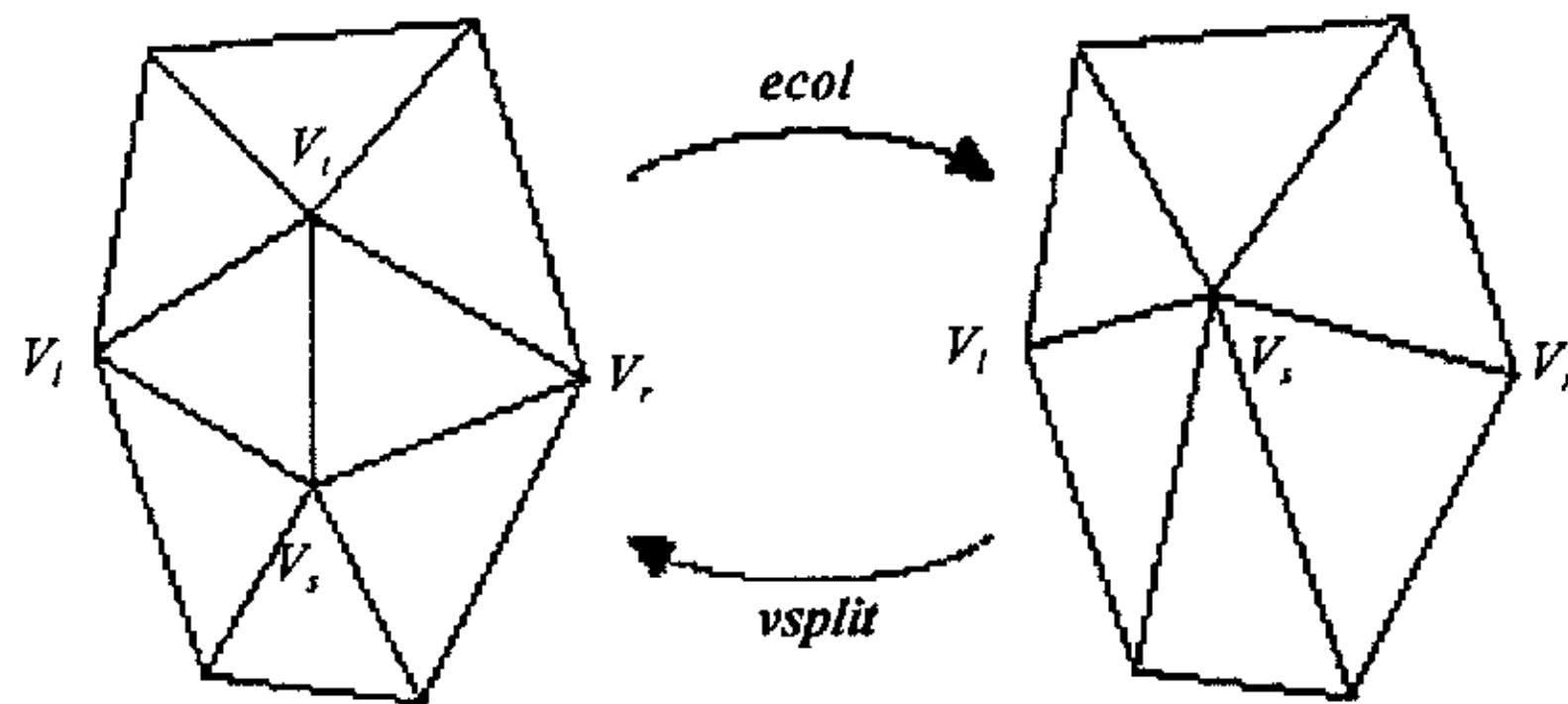


图 1.4 边删除及边分裂操作

Hoppe在1996年提出基于最小能量函数法建立渐进网格模型[Hoppe96]。算法首先通过递归的边删除（Edge Collapse）操作求得简化模型。如图1.4所示，边 $\{V_b, V_s\}$ 的删除操作 $ecol(\{V_b, V_s\})$ 把两个点 V_l 、 V_s 合并为一点 V_s 。该删除操作使顶点 V_l 以及两相邻面片 $\{V_s, V_b, V_r\}$ 、 $\{V_b, V_s, V_r\}$ 从网格模型中被删除。原始模型 $\hat{M} = M^n$ ，通过 n 次连续的边删除操作简化为较粗的近似模型 M^0 ：

$$(\hat{M} = M^n) \xrightarrow{ecol_{n-1}} \dots \xrightarrow{ecol_1} M^1 \xrightarrow{ecol_0} M^0。$$

Hoppe通过从 M^n 到 M^0 的各边删除运算的逆运算——顶点分裂（Vertex Split—

$vsplit$), 保存原两顶点 V_i 、 V_j 的位置信息、顶点所在局部区域的连接关系、以及相关面片、顶点的离散属性和标量属性等信息。从而, 任意三角形网格 \hat{M} 可以表示为一个简化模型 M^0 以及一系列顶点分裂操作 ($vsplit$):

$$M^0 \xrightarrow{vsplit_0} M^1 \xrightarrow{vsplit_1} \dots \xrightarrow{vsplit_{n-1}} (M^n = \hat{M}),$$

即为 $(M^0, \{vsplit_0, \dots, vsplit_{n-1}\})$ 。其中, M^0 为基网格 (Base Mesh)。Hoppe 把这种模型的表示形式定义为渐进网格 (Progressive Mesh——PM)。这样, 如果给定模型的渐进网格, 我们可以通过对基网格施行系列顶点分裂操作, 得到任意分辨率的简化模型。

PM 的思想与基于小波分解方法有些类似, 二者都保存原始模型的一个基本简化模型和一系列细节记录, 从而建立原始模型的连续多分辨率表示。但 PM 在很多方面克服了小波分解方法的局限性, 具有更好的简化效果以及更广泛的应用。如 PM 没有对模型的分割连通性的要求, 可以完全恢复原始模型。此外, PM 可以在任一细节层次模型上引入特征边 (Crease), 破除了小波的方法只能恢复存在于基网格模型上的特征边的限制。PM 方法还能在简化过程中提取连续、分段连续或离散的表面属性。另一方面, 小波方法在模型的多分辨率编辑 (Multiresolution Mesh Editing)、控制近似模型的最大误差、对模型几何、颜色等信息分别独立压缩等几个方面存在优势。

Liu[Liu98]、Cheng[Cheng98]等分别尝试了利用顶点删除和三角形删除建立渐进网格模型的方法。

1.3.2 粗化的方法 (Decimation Method)

粗化算法在对一般模型的简化中应用比较广泛, 许多研究者在这方面做了大量的工作, 提出了很多相关算法。

- 基于顶点聚合的算法(Vertex Clustering):

顶点聚合算法[Rossignac93, Low97, Schaufler95, Luebke96]通过空间划分把模型中的顶点分成顶点簇, 然后把同簇内的点聚合成为一点。算法的简化操作不依赖于模型的拓扑信息 (邻接关系), 而只与几何信息 (顶点坐标) 有关。

最简单的聚合算法是 Rossignac 和 Borrel 提出的均匀顶点聚合算法 [Rossignac93]。算法首先建立整个模型的包围盒，并对包围盒进行均匀划分，同时相应地把模型顶点分到各个网格单元，在每一网格中选择可以代表整个网格单元中所有顶点的优化（“Optimal”）点作为聚合顶点（如距离单元内所有顶点的加权平均位置最近的顶点，或使误差值最小的点）。

该算法执行的效率很高，并且可以改变模型的拓扑结构，特别适用于通常会包含细小几何特征模型的机械CAD模型。但由算法基本原理可知，简化结果的好坏在很大程度上依赖于网格单元的分布。同时，它不能简化大于网格单元尺寸的几何特征。假设一矩形平面分割为三角网格，其中所有三角形的尺寸都大于网格单元，则用这一方法不能对该网格模型实施任何简化操作。而众所周知，该矩形平面实际上可以无误差地简化为两个三角形。

显然，建立更有效的空间剖分策略可以使均匀聚合算法得到改善。Luebke 通过建立空间的适应性八叉树剖分对算法进行了改进 [Luebke96, Luebke97]。Low 和 Tan 则提出采用立方体、球等任意简单的形状作为空间剖分单元，并将单元的中心定位于单元内重要度最高的顶点 [Low97]。对于同时被包含在多个单元内的顶点，算法根据其与各单元中心距离的远近来进行分配。

基于顶点聚合的算法通常运行速度很快，而且可以对任意三角形网格模型进行简化。它可支持非流形体的简化，而且不需要完整的网格拓扑连结信息。该算法非常适用于原始模型的采样高度密集且所要求简化程度不高的情况。当网格模型三角面片的尺寸小于空间分割单元尺寸时，算法的执行结果较好。简化过程中，网格的所有顶点只在网格单元内移动，即所有网格顶点简化后的所移动的距离均不大于网格单元的对角线。因此，由聚合算法生成简化模型的 Hausdorff 近似误差都有确定的上界。但是，为了实现对模型的大幅度简化，网格单元的尺寸迅速增大，从而会导致过大的误差。因此，在简化程度较大的情况下，所生成模型的质量会迅速降低。

- 基于区域合并的算法(Region Merging)

区域合并算法 [Deheamer91, Calvin91, Calvin94, Calvin96, Hinker93,

Gourdon95, Reddy96]通过合并网格模型的表面区域进行简化。Kalvin和Taylor[Kalvin94, Kalvin96]的“超面 (Superface)”算法基于共面准则把整个网格模型分割成连通区域 (Superface), 分别用多边形面片代替各个区域, 并对多边形面片的边界进行简化, 然后对多边形面片重新三角化。Superface算法的简化结果则较好, 并且可以有效地控制近似误差。但该算法通常只限于流形体模型的简化, 并且不会改变模型的拓扑结构。Hinker和Hanson[Hinker93]以及Gourdon[Gourdon95]则提出了较适于处理曲率变化不大的光滑表面模型的算法。

与其它多分辨率模型简化方法相比, 区域合并方法虽然在保持模型几何特征方面具有无法比拟的优越性, 但其算法实现比较复杂。因此, 人们在这方面所做的研究较少。本文所提出的基于高斯球的近似共面合并多层次细节模型 (第二章) 通过对高斯球面进行层次分割, 简洁、有效地实现了近似共面区域合并进而建立多层次细节模型[Cao99]。

● 基于顶点删除的算法 (Vertex Decimation)

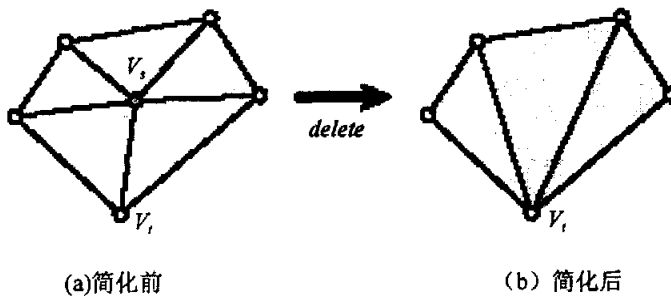


图 1.5 点删除操作 (删除点 V_i , 并对所余空洞重新三角化)

基于顶点删除的算法, 是应用较为广泛的模型简化算法之一。它通过删除满足误差标准的顶点, 并对由此产生的空洞进行三角剖分生成简化的模型 (如图1.5) [Schroeder92, Schroeder97, Soucy96a, Klein96, Ciampalini97, Kobbelt98, Zhou96, Li98b, Turk92]。

Turk和Schroeder都在1992年把顶点删除的策略应用到各自的模型简化算法中 [Turk92, Schroeder92]。

Turk根据曲率将给定数量的一组新点分布到模型的曲面上，新顶点与老顶点结合形成中间网格，然后用顶点删除操作去除老顶点，对遗留的空洞重新三角化形成简化网格[Turk92]。

Schroeder也提出了基于顶点删除的简化算法[Schroeder92]，通过删除顶点去除模型中的部分细节，降低模型复杂度。该算法首先根据点的局部拓扑、几何信息将各顶点分类（简单点、复杂点、边界点、内部点、角点）。在对模型进行简化的过程中，选择所要删除的顶点，然后把所有与该点相邻的面片全部删除，并对所产生的空洞重新三角化。由于这里的重新三角化算法需要把局部模型曲面投影到平面上，所以该简化算法只能应用于流形曲面。并且，算法的基本简化操作——点删除也不能用于实现拓扑的简化。1997年，Schroeder通过执行剪切（Cutting）、缝合（Stitching）等操作，消除了这一算法在流形体及拓扑简化等方面的限制[Schroeder97]。在较早的顶点删除算法[Schroeder92]中，对近似误差的估计十分保守（点到平均平面或点到边的距离），而以后的算法则采用了更为精确的误差度量方法，如局部Hausdorff误差等。算法记录原始模型的顶点与近似模型上相应面片之间的对应关系，它们之间的距离则定义了模型的近似误差[Klein96]。

Schroeder的基于点删除的模型简化算法在时间及空间上都有较高的效率，但它很难使模型表面网格保持光滑[Schroeder97]，原本表面平滑的网格模型在简化过程中会变得非常粗糙。其它的顶点删除算法可以生成较高质量的近似模型，但算法执行速度较慢，而且耗费大量的内存空间。

顶点删除算法实际上也是一种递归折叠（Iterative Contraction）算法。在特殊情况下，删除点(V)可以很容易地通过折叠边(e)而实现。如图1.5中删除点 V_3 可以很容易地通过边折叠（删除边 (V_3, V_1) ，以 V_1 作为新点）实现。利用边折叠的方法删除顶点[Schroeder97]，不需要去找可以避免产生边交叉的投影平面，因而[Schroeder92]具有更高的算法鲁棒性。

- 基于递归折叠的算法（Iterative Contraction）：

还有一种多分辨率模型采用基于边的递归折叠的简化算法[Hoppe96,

Popović97, Hoppe98, Ronfard96a, Guéziec95, Guéziec96, Garland97, Gieng98, Algorri96, Lindstorm98, Melax98, Lau98, Cao99]。本文在第三章和第四章所提出的算法均属于这一类算法。算法首先根据边删除后引起的误差对边进行排序，引起误差最小的边优先级最高，最先简化。若一条边被折叠，则其两端点聚合为一点，同时相关三角形退化为边，并加以删除（如图1.6）。递归删除优先级最高的

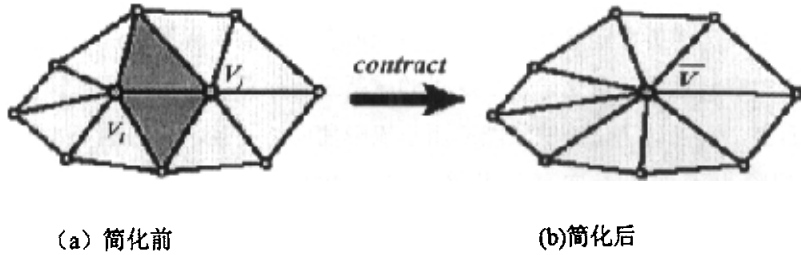


图 1.6 边折叠操作（删除边 (V_i, V_j) ，则两端点合并为一点 \bar{V} ；以较深阴影标识的三角形退化为边，并被删除。）

边，可生成不断简化的模型。如果算法不显式地保持拓扑，边折叠算法可以实现对模型拓扑的改变（如将模型表面的孔封闭等）。

最先提出并实现基于边折叠的简化机制的是Hoppe等人[Hoppe93]。后来，Garland[Garland97]和Popović等[Popović97]把边折叠扩展到任意点对的折叠，从而可以把分离的两部分网格模型简化成为连通的整体，实现拓扑简化。由于递归折叠算法的基本操作并不要求待简化的局部区域为流形，可用于任意模型的简化，因此，与点删除的算法不同，基于折叠的算法可以很方便地应用于非流形曲面模型的简化。

前面提到的Hoppe所提出的基于渐进网格模型[Hoppe96]建立简化模型(Base Approximation)的算法实际上也是最早且较为成功的一种递归折叠简化算法。它根据能量函数的变化情况决定边折叠的顺序。其能量函数包括 E_{dist} 、 E_{spring} 、 E_{scalar} 、 E_{disc} 等四部分：

$$E(M) = E_{dist}(M) + E_{spring}(M) + E_{scalar}(M) + E_{disc}(M)。$$

其中， M 为简化模型； E_{dist} 是距离能量（Distance Energy），为原始模型上的各顶点

到 M 的距离平方和，表示 M 对原始模型的近似程度； E_{spring} 为弹性能量（Spring Energy），即设模型每一边的初始能量为0，并设定各边的弹性系数，由此所求得 M 中各边能量之和； E_{scalar} 用于度量模型表面的标量属性（Scalar Attributes）误差——漫射颜色（Diffuse Color）、法向（Normal）、纹理坐标（Texture Coordinates）； E_{disc} 则度量简化所引起的标量属性不连续曲线（Discontinuity Curves）的几何误差。该算法所生成的简化模型具有很高的质量，但是其执行时间很长，执行对一70,000个面片模型的简化需要大约一个小时的时间[Hoppe97]。算法可以执行诸如封闭曲面上的孔等拓扑简化过程，之后Popović和Hoppe对算法进行了改善，使其可以连通两个分离区域[Popović97]。Hoppe等在1993年提出的网格优化（Mesh Optimization）算法[Hoppe93]是建立渐进网格模型的基础[Hoppe96]。算法在简化过程中总是根据能量的变化查找最优的简化步骤，采用边折叠、边分裂、边翻转等简化操作。该算法在时间上的开销比[Hoppe96]更高，但可以得到最好的简化效果。

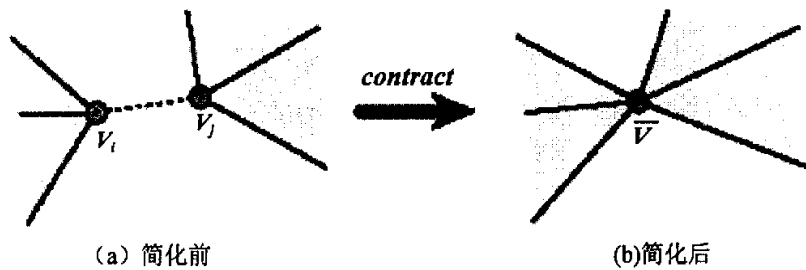


图 1.7 点对折叠操作（点对 (V_i, V_j) 合并为一点 \bar{V} ，使两个分离区域连接起来，没有任何面片的退化及删除。）

Ronfard 和 Rossignac 提出的简化算法执行效率较高[Ronfard96a]。简化过程中，模型的每一顶点都对应一平面集合，该点的误差即为与这些平面距离的最大值的平方。当边折叠操作使两点合并为一点时，该新点的面片集合为原两点面片集合的并集。显然，这种误差度量方法与Hoppe的方法相比，大大节省了在时间上的开销，Ronfard和Rossignac指出，可以由该误差度量得到局部Hausdorff误差[Ronfard96b]。该算法也具有较好的简化结果，而且执行效率比很多其它的算法高。

Garland提出的基于二次误差的简化算法把简化操作从对边(edges)的递归删除(如图1.6)扩展到点对(vertex pairs)的递归删除(如图1.7)。其所采用的误差度量方法与Rongard和Rossignac[Ronfard96a]的方法类似[Garland97],即以二次误差(quadric error metrics)度量各简化操作引起的误差(cost)。设 V 为模型上一顶点,与之相关有一平面集合。 V 到集合中各平面距离的平方和,定义为该顶点的误差:

$$\Delta(V) = \begin{pmatrix} v_x & v_y & v_z \end{pmatrix}^T = \sum_{p \in \text{planes}(v)} (p^T v)^2 = v^T \left(\sum_{p \in \text{planes}(v)} K_p \right) v = v^T Q v \quad .$$

其中, $p = [a \ b \ c \ d]^T$, 表示与该顶点相关的平面; $K_p = pp^T$, 用于计算点到平面 p 的距离的平方值; Q 表示点 V 的误差矩阵,为平面集合中所有平面的 K_p 之和,即 $Q = \sum_{p \in \text{planes}(v)} K_p$ 。点相关平面集合的初始值设为以其为顶点的所有三角形。根据各顶点初始相关平面集合,算法首先求得各顶点的初始矩阵 Q ,并确定二次误差最小的点为有效点对的折叠目标点(新点)。新点的误差矩阵为原点对中两点的误差矩阵(Q_1, Q_2)之和,即 $\bar{Q} = (Q_1 + Q_2)$,而该点的二次误差即为该点对的简化误差(cost)。Garland的方法在简化过程中运用简单的矩阵加法进行误差矩阵的传递,因此算法执行效率很高。在一定程度上实现了在简化效率和质量之间的折衷,而且算法简单,容易实现。

Lindstrom 和 Turk 在1998年提出“节省内存(Memoryless)”的算法[Lindstorm98]。该方法试图通过无需参考原始模型的局部边删除操作生成简化模型。由于在简化过程中不必保持原始模型的信息,节省了存储空间。算法以边删除操作引起的模型局部变化(由简化引起的相邻三角形旋转)作为简化操作的误差,依据保持模型总体体积不变(相邻三角形旋转所扫成四面体的体积之和为0)以及使单个相邻三角形旋转所形成的四面体体积最小的原则确定边的删除顺序以及生成新顶点的位置[Lindstorm98]。[Lindstorm98]指出,该算法除了节省存储空间之外,在时间上的开销也较小,而且简化模型的质量较高。

递归边折叠算法的一个主要优点在于其在简化过程中自然形成的层次结构。我们可以很方便地由这种层次结构生成模型的多分辨率表示[Hoppe96, Hoppe97, Hoppe98, Xia96]。

除了基于边的递归折叠算法，Hamman 等还提出了基于三角形删除的多分辨率模型[Hamman94]。

- 基于三角形折叠的简化算法 (Triangle Contraction)

三角形折叠算法通过将三角形的三个顶点合并为一点，并删除退化的相邻三角形以及原三角形对三角网格模型进行简化 [Hamman94, Gieng97, Ma98, Zhou98]。一次三角形折叠操作相当于三次边折叠操作，因此三角形折叠算法运算速度较高，但同时也使其只能对简化操作进行较粗的控制。

显然，我们还可以将删除操作进一步扩展到网格模型上更大的连通区域。

1.3.3 考虑材料属性 (Material Properties) 的多分辨率模型

现有的多分辨率模型简化算法大都着重于对模型几何属性的保持。但在实际应用中，除几何信息之外，场景模型还包含一些表面属性信息，如 Material Identifier、表面法向 (n_x, n_y, n_z) 以及颜色 (r, g, b) 、纹理坐标 (u, v) 等材料属性。这些表面属性决定了绘制模型中面片的 Shader Function [Upstill90]。对于象高度场这样特定类型的模型，用较简单的方法进行处理 [Garland95] (如在模型顶点采样近似表示地形颜色) 就可以获得相当好的结果。但对于一般的模型，需要更复杂的处理技术 [Hoppe96]。

Hoppe [Hoppe96] 在误差度量函数中除了反映几何属性误差项之外，还包含有标量属性误差项，以及标量属性不连续曲线误差，同时反映了各顶点的标量属性误差以及各面片的属性误差。算法可以很有效地控制模型的材料属性误差，但需要很高的时间开销。Certain 等人则用小波保持表面颜色属性 [Certain96]。Hughes 等人研究了具有颜色属性的网格模型的辐射度全局光照模型的简化 [Hughes96]。他利用对数函数把顶点的颜色属性转换到线性空间，然后再进行简化。Garland 在 [Garland98] 中对 [Garland97] 基于二次误差的模型简化算法进行了改进，把二次形误差增加5维 (3 维表示RGB颜色，2维表示纹理坐标)，从而把对材料属性误差的控制添加到简化过程中。

Cohen [Cohen98, Cohen97]、Maruya [Maruya95]、Soucy [Soucy96b]、和

Cognoni[Cognoni98]等通过建立材料属性到模型表面上的映射, 并对该映射进行重新参数化的方法在简化模型中添加对材料属性的考虑。

在近些年中, 多分辨率模型的简化算法研究方面取得了非常重要的进展, 并已经产生了很多非常有效的算法。一些商业软件包中已经出现了模型简化工具, 并可由简化生成有效的多分辨率模型[Hoppe97, Luebke97, Xia96, Floriani98]。Hoppe的简化算法可以得到非常好的简化效果, 但其速度很慢。顶点聚合算法[Rossignac93]运行速度很快, 并可以广泛地应用于任意三角片模型, 但其简化效果不尽人意。而Schroeder等人提出的顶点删除算法[Schroeder92, Schroeder97]执行相当快捷, 但它不能保持模型表面的平滑。而Garland的算法则实现了高质量的简化效果以及较高的运行速度之间的折衷[Garland97]。

本文利用层次分割的Gauss球面基于共面合并的原则简单快捷地建立多分辨率模型(第二章), 有效地保持了模型的几何特征; 此外, 考虑到在大多数情况下, 网格模型本身实际上是对曲面场景模型的近似, 本文分别通过分割插值(第三章)以及隐函数插值(第四章)建立多分辨率模型, 成功地减小了对原始模型的近似误差。

1.4 对简化模型质量的评价

在1.3节已经提到, 建立多分辨率模型的目的是为了生成尽可能与原始模型近似的简化模型。因此, 简化模型的近似质量是评价简化算法的一个重要指标。

为了评估近似质量, 我们需要给“近似”的概念加以量化的定义。设给定一网格模型 M , M' 为其近似简化模型, $E(M, M')$ 表示近似模型 M' 的近似误差。 E 的值越小表示该近似模型越接近于原始模型 M 。显然, 如何计算 E 即如何评判模型的近似程度与简化模型的应用领域密切相关。

● 视觉近似误差

对于多分辨率模型技术研究的主要应用领域之一——绘制系统来说, 视觉上的近似程度(Similarity of Appearance)等价当然应成为评价简化模型近似质量的

最终标准[Heckbert94]。在场景的绘制过程中，如果用 M^* 代替 M ，所引起的视觉误差不仅决定于二者的形状，还与当前的观察状态 ξ （视线方向、视距大小等）有关。设在观察状态 ξ 下， M 和 M^* 的图象分别为 I_ξ 和 I_ξ^* ，则当 I_ξ 和 I_ξ^* 相同时，模型 M 和 M^* 视觉等价，而二者之间的差异则代表模型 M 和 M^* 在视觉上的近似程度。若 I_ξ 和 I_ξ^* 均为 3×3 的RGB光栅图形，我们可以把 I_ξ^* 相对于 I_ξ 的误差做如下定义[Garland99]：

$$\|I_\xi - I_\xi^*\|_{img} = \frac{1}{m^2} \sum_u \sum_v \|I_\xi(u, v) - I_\xi^*(u, v)\|^2。$$

其中 $\|I_\xi(u, v) - I_\xi^*(u, v)\|$ 表示两RGB矢量之差的欧几里德长度。 M^* 与 M 在视觉上的近似程度越高， $\|I_\xi - I_\xi^*\|_{img}$ 的值越小。用这种方法我们可以求任意视觉状态下原始模型与简化模型之间的视觉误差。

采用基于视觉等价的误差度量方法的主要优点在于可以直接度量模型在视觉上的近似程度，而这正是绘制系统所关心的问题。此外，该方法还允许简化算法剔除不可见场景。例如一个包含一陈设复杂的房间的场景，如果视点的移动范围只限于房间之外而不会进入房间，根据视觉等价原则，可以把房间内的场景全部删除而不会使误差值 E 增大。虽然基于视觉等价的误差度量方法应用于模型简化有很多诱人的优越之处，但也存在一些问题。最重要的问题就是它需要对可能的视点进行充分的采样。如果在采样中漏掉了视点空间的重要部分，可能导致一些非常重要的视觉特征被删除。同时，由于每一视点的采样实际上都需要很大开销，因此采样视点的数目会受到限制。正是这种无法解决的矛盾，使多分辨率模型的研究人员一般不采用基于视觉等价的误差度量原则，而采用较易实现的几何误差度量方法，即用几何误差来近似代替视觉误差，以形状（Shape）上的近似程度作为评定近似质量的标准。

● 几何近似误差（Geometric Approximation Error）

众所周知， L_∞ 和 L_2 范数[Prenter76]是最为常见的函数误差。设函数 $f(t)$ 为一浮点函数， $g(t)$ 为 $f(t)$ 的近似函数，它们的定义域为 $[a, b]$ 。则 L_∞ 范数用于度量 $g(t)$ 与 $f(t)$ 的之间的最大偏差，定义为

$$\|f - g\|_\infty = \max_{a \leq t \leq b} |f(t) - g(t)|；$$

L_2 范数用于度量 $g(t)$ 与 $f(t)$ 之间的平均偏差, 定义为

$$\|f - g\|_2 = \sqrt{\int_a^b (f(t) - g(t))^2 dt}.$$

如果函数 $g(t)$ 相对于与 $f(t)$ 的误差最小, 则 $g(t)$ 被称为 $f(t)$ 的最优近似函数。

L_∞ 范数通常被作为评价近似函数的较严格的误差尺度。由于它直接反映了原始函数与近似函数之间的全局最大距离, 因此可以很方便地控制近似误差, 保证近似质量。但是, L_∞ 范数也存在对噪声及局部偏差过度敏感等缺点。另一方面, L_2 范数可以较好地反映函数在整体上的匹配程度, 但可能忽略较大局部偏差。

显然, 我们可以沿用 L_∞ 和 L_2 范数等简捷的函数近似误差来表达网格模型的几何近似误差。给定网格模型 M , $P(M)$ 为 M 上的所有点的集合, 则点 V 到 M 的距离定义为 M 上 V 的最近点与 V 之间的距离:

$$d_v(M) = \min_{w \in P(M)} \|v - w\|.$$

其中, $\|\cdot\|$ 为欧几里德空间矢量长度算子。

Hausdorff距离 $\hat{E}_{\max}(M_1, M_2)$ [Preparata85]是与 L_∞ 范数相对应的常用几何误差度量, 用于定义模型之间的最大偏差,

$$\hat{E}_{\max}(M_1, M_2) = \max\left(\max_{v \in P(M_1)} d_v(M_2), \max_{v \in P(M_2)} d_v(M_1)\right).$$

其中, $d_v(M)$ 表示点 V 到模型 M 的距离。由Hausdorff误差的定义可知, 若 M^* 为模型 M 的近似模型, Hausdorff误差 $\hat{E}_{\max}(M, M^*) \leq a$, 则表示近似模型 M^* 上的任一点到原始模型 M 的距离都不超过 a , 并且 M 上的任一点到 M^* 的距离也都不超过 a 。

同理, 可以定义与 L_2 范数相对应的几何误差度量 $\hat{E}_{\text{avg}}(M_1, M_2)$, 表示两模型之间距离的二次平均值, 如下式所示,

$$\hat{E}_{\text{avg}}(M_1, M_2) = \frac{1}{W_1 + W_2} \left(\int_{P(M_1)} d_v^2(M_2) dv + \int_{P(M_2)} d_v^2(M_1) dv \right).$$

其中, W_1 、 W_2 分别表示模型 M_1 、 M_2 的表面面积。

在实际应用中, 精确计算 $\hat{E}_{\max}(M_1, M_2)$ 或 $\hat{E}_{\text{avg}}(M_1, M_2)$ 误差不仅要计算两个模型上所有顶点之间的距离, 还要考虑位于各面片内部的点, 因此计算量非常大,

需要极大的开销。由于三角形全部为凸，如果三角形T的各顶点移动的距离都不超过D，则其内部点也不会超过D。由此可知，简化操作引起的简化模型上点的偏移量不会超过模型顶点的偏移量。因此，人们通常用离散点的集合即模型所有顶点来代替整个模型上的点以求得近似误差值，即以离散的Hausdorff距离检验模型的近似质量。Hausdorff距离小则表明简化模型与原始模型的近似程度较高。分别给定网格 M_1 、 M_2 的点集 $P(M_1)$ 及 $P(M_2)$ 。集合 X_1 、 X_2 则分别为 $P(M_1)$ 及 $P(M_2)$ 的子集并分别包含 k_1 、 k_2 个顶点。它们至少包含对应模型所有顶点。这样，模型的近似误差 $\hat{E}_{\max}(M_1, M_2)$ 和 $\hat{E}_{\text{avg}}(M_1, M_2)$ 可分别近似表示为

$$E_{\max}(M_1, M_2) = \max\left(\max_{v \in X_1} d_v(M_2), \max_{v \in X_2} d_v(M_1)\right);$$

$$E_{\text{avg}}(M_1, M_2) = \frac{1}{k_1 + k_2} \left(\sum_{v \in X_1} d_v^2(M_2) + \sum_{v \in X_2} d_v^2(M_1) \right).$$

正如函数的 L_2 范数， $E_{\text{avg}}(M_1, M_2)$ 可以很好地反映两个模型在整体上的匹配程度；而 $E_{\max}(M_1, M_2)$ 则如函数的 L_∞ 范数，虽然对噪声及局部偏差较为敏感，但是它直接反映了原始模型与近似模型之间的全局最大距离，能较好地反映简化模型相对于原始模型在几何上的近似程度，因此可以作为对简化算法进行评价的标准之一。另一方面，由于即使是离散的Hausdorff距离的计算也需要很大的开销，因此，简化算法一般都不直接采用它作为简化过程中的误差度量方式，而采用一些更简单、时间复杂度低的度量（本文3.3节）。

● 网格质量 (Mesh Quality)

除了考虑简化模型在几何结构方面对原始模型的保真程度，场景的网格模型的质量还涉及网格模型中面片的质量。特别是三角网格模型，三角面片的形状也是网格质量非常重要的一个方面。在一些特殊的研究领域如有限元中，三角面片的形状是被特别关注的问题。狭长三角形 (Sliver Triangles——包含接近0的较小内角的三角形) 会使近似模型的质量降低 [Babuska76]，甚至影响有限元分析结果，在求解过程中引起数值振荡 [Prenter75]。Delaunay 三角化是目前使用较为广泛的一种优化三角化方法，它使所有三角形的最小内角最大，以生成高质量网格 [Preparata85, Chew93]。但是，带有狭长三角形的网格模型却具有较小的 L_2 或 L_∞ 范数误差 [Rippa92]。

在本文第三章所提出的算法中，我们的简化过程控制函数即同时包含了反映模型几何误差以及三角形形状的误差项，采取了在模型简化误差以及网格质量之间折衷的简化准则。

1.5 论文工作

本论文的工作主要体现在以下几个方面：

综述了多分辨率模型方法到目前为止所取得的研究成果，分析了各种方法的特点以及相互之间的关系。

提出了一个基于高斯球的近似共面合并多层次细节模型算法。它基于对高斯球面的分割，快速建立网格模型的近似共面集合，生成超面，并通过对超面的三角化生成原始模型的基本简化模型。算法还建立了高斯球面网格的层次结构，并基于这一层次结构建立了近似共面合并的多分辨率模型。算法的运行速度快，并且能有效保持模型的几何特征。

提出了一个基于分割插值的连续多分辨率模型。算法通过递归地删除网格模型中的边得到连续的简化模型。建立可调加权控制函数来控制边的简化顺序，并在模型简化的不同阶段，通过调节控制函数的权值执行不同的简化原则，使得视觉重要度低的边优先删除。算法基于分割插值造型的原理，用临近点插值模板求取边简化生成的新点，使简化模型保持较小的近似误差。

提出了一个基于隐函数插值的连续多分辨率模型。该算法通过对原始网格模型分区域变分插值，生成对应的隐函数曲面，并把对该曲面的重新采样作为简化的新点。在简化过程中，区域变分插值曲面保持不便，从而避免了累积误差。

1.6 小结

这一章介绍了多分辨率模型技术产生的背景以及研究快速绘制的目的及意义；分析比较了现有的多分辨率模型算法；讨论了简化模型的近似质量的评估方法；最后简要介绍了论文所做的工作。在后面几章，我们将对本文在多分辨率模型方面的工作进行详细讨论。

论文其余各章的内容如下：

第二章介绍基于高斯球的近似共面合并层次细节模型；

第三章介绍基于分割插值的连续多分辨率模型；

第四章介绍基于隐函数插值的连续多分辨率模型；

第五章总结论文工作，及展望今后多分辨率模型技术的发展方向。

第二章 基于近似共面合并的多分辨率模型

目前,已有的多分辨率模型的简化算法大都是根据景物网格模型结点之间,或点与面片之间,或点与边之间的空间距离来判断是否保留几何元素。在绘制过程中,通过控制误差阈值,使简化后的网格模型与原模型在屏幕空间的投影误差足够小,以保持简化模型与原模型视觉等价。注意到景物模型中,有些几何结构在景物空间的几何尺寸较小,但对定义景物的几何特征有很重要的作用。如果单纯以景物空间的距离作为判据,在取较大阈值时,势必会造成这些景物几何特征的丢失。因而,为保持视觉等价,需要定义较小的阈值。但这样生成的景物简化模型,并不能达到充分的简化程度。

众所周知,景物模型的几何特征与模型表面的法向变化有着严格的对应关系。模型的特征边即为模型表面法向变化剧烈而形成的棱边,特征点则为两条以上特征边的交点。基于景物网格模型中面片的共面特征进行简化,可以在有效地保持模型的特征边、特征点等重要几何特征的前提下,最大程度地简化景物模型。另一方面,基于近似共面合并的算法的时间复杂度较低,可以实现复杂模型的快速简化。

本章提出了一种基于高斯球的共面合并自动生成场景多层次细节模型的算法。将一单位球定义为高斯球,并按一定的规则对其进行层次分割。然后根据各景物面片的法向在高斯球上的分布,将具有近似一致法向的面片划分为面片簇;在面片簇内合并相邻面片,生成近似共面区域;然后移去其内部点,对区域边界重新三角化。算法还结合了临近边界点合并等原则,删除对表达场景几何特征重要度低的顶点。算法利用平面九叉树进行合并操作,生成具有多层次细节的场景模型。该算法计算简单,时间复杂度低,可以很好地保持模型的几何特征,且与应用领域无关。

本章将在综述前人在共面合并多层次细节模型方面所做的工作的同时,对基

于高斯球的近似共面合并多分辨率模型加以详细介绍。

2.1 共面合并多层次细节模型

共面合并多层次细节模型的简化算法是一种由共面检测驱动的区域合并算法，它把模型中共面或近似共面的面片合并，形成较大的多边形，然后对该多边形重新三角化，得到较简化的网格模型[DeHaemer91, Hinker93, Kalvin91, Kalvin96, Gourdon95, Reddy96]。共面合并的算法在保持模型几何特征方面有很高的优越性。此外，算法在执行效率上也具有一定的优势。

1991年，Dehaemer为了对Schmitt建立采样点拟合曲面的算法[Schmitt86]进行优化，提出以通过曲率最高点的经线和纬线对曲面进行划分，从而生成保持曲面近似共面区域的多边形网格[Dehaemer91]。同年，Kalvin基于四个矩形相交于一点则四个矩形共面（a Tile）或为两个相交的平面（a Crease）的原理，在他的曲面构造算法ALLIGATOR中，通过查找由B-rep法表示的3D医学数据中同时位于四个矩形上的顶点，查找共面区域，并将它们合并为一个平面，在保持曲面的几何形状的同时消除了曲面构造中的通常存在的大量冗余的共面多边形，大大减少了数据量[Kalvin91]。Dehaemer和Kalvin的简化算法只是体现了共面合并算法的一些思想，Dehaemer的算法显然很粗糙，只是通过在高曲率点进行分割，使近似共面的区域得到保持，并没有真正地检测面片的共面性；Kalvin的算法只对特定情况下（四个相交于一点的矩形）的共面区域进行合并，没有建立普遍意义上的共面合并准则。

后来，Hinker[Hinker93]和Kalvin[Kalvin94, Kalvin96]分别提出了基于精确的共面准则不依赖于应用的多边形网格的模型简化算法——几何优化（Mesh Optimization）超面（Superface）的算法。这两种方法与本文所提出的基于Gauss球的近似共面合并算法类似，都是通过合并一组相邻的近似共面面片，形成多边形区域（称为超面，Superface），然后对区域边界重新三角化来生成原始模型的简化模型。

但三者在建建立近似共面面片集合的方法、共面标准、生成共面区域边界以及超面的三角化方法等很多方面不尽相同。

- 建立近似共面面片集合及共面标准

Salesin提出基于代表树 (Representative Tree) 建立近似共面集合 [Salesin92]。算法通过递归地执行近似排序建立链表结构, 把以一定的误差 ϵ (法向夹角, 及面片之间的距离误差) 近似共面的面片存于同一节点, 从而实现对面片的分类, 建立了近似共面面片集合。其中, 算法把第一个进入节点的面片的法向作为判断近似共面的代表法向 (Representative Normal)。显然, 节点内面片的法向夹角最大为 2ϵ 。由于共面集合的代表法向的随机性, 可能导致不优化的近似共面分类结果。代表树算法的时间复杂度为 $O(n \log n)$, 其中, n 为原始模型中的面片数。Hinker 对 Salesin 的代表法向的定义加以改进, 其初值的确定与 Salesin 相同, 而当节点包含面片数目多于一个, 则以节点内所有面片的法向的平均值作为该节点新的代表法向。改进的算法使与节点内面片法向的最大夹角不超过阈值的面片都加入节点面片集, 从而建立分配更为合理的近似共面集合。同时, Hinker 的算法保持了代表树算法的复杂度 [Hinker93]。

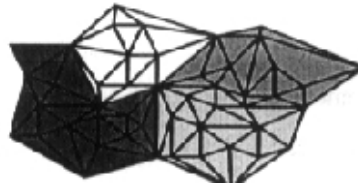


图 2.1 曲面片 (Surface Patch) 和超面 (Superface)

Kalvin 的算法基于以下定义:

曲面片 (Surface Patch): 是一些连通的多边形面片集合 (如图 2.1 所示, 用同一灰度标识的三角形分别为连通三角形集合, 组成四个曲面片);

曲面片的边界 (Boundary): 位于其周边的顶点集合;

超面 (Superface): 由曲面片的边界所围形成的非平面多边形 (图 2.1 中用同一灰度标识的区域称为超面);

边界面片 (Border Faces): 曲面片周围的面片 (与曲面片相邻的面片) 称为

对应超面的边界面片。

Kalvin通过把原始模型分割为超面建立共面区域[Kalvin96]。超面是从一个初始的“种子(Seed)”面片开始,并基于一种贪婪的迭代算法增长而生成的。若边界面片满足几何距离误差检测标准(若经扩展,原始网格中的各结点与扩展后形成的简化网格的距离小于给定的阈值 ε ,则满足标准。)则被合并到相应的超面,实现超面的扩展;当没有可以合并的边界面片时,该超面停止增长。而当原始模型上的所有面片全部都合并到超面中时,原始模型的分割过程结束。此时,原始模型被分割成很多共面区域 [Kalvin96]。超面算法把原始模型分割成超面的时间复杂度为 $O(n)$,其中 n 为原始模型中的面片数。

以上两种方法都以一定的共面准则逐一判断各面片可否加入共面面片集合,算法复杂度较高。本章则基于高斯球面的网格划分,按照面片的法向将其分配到相应的球面网格,从而形成近似共面面片集合(见本文2.3.1)。本算法由高斯球面网格来限制近似共面面片集合内面片之间的最大夹角,而不需要求取待分配面片与已分配面片之间的关系,从而显著简化了算法,提高了算法的执行效率。

- 生成共面区域边界

Hinker的优化算法对共面集合内的边根据端点坐标排序(三坐标的优先级顺序依次为X坐标、Y坐标、Z坐标。),建立边表(Segment List)。删除其中的内部边以及重叠边等非边界边,然后对边表中所剩余的边界边依据端点坐标进行递归的二叉树搜索查找共点边,直到边表为空,则形成封闭的多边形,即近似共面区域边界。而本章的算法,由于建立了层次分割的高斯球面网格,以及对应于各球面网格的三角面片组,因此我们可以很方便地辨别边界点,并根据点的邻接关系搜索共面区域边界(本文2.3.2)。

- 共面区域的三角化

Hinker采用简单的Kong切耳算法[Kong90],以多边形边界上三个相邻顶点所组成的三角形作为剖分三角形,并在确定一个剖分三角形后,对所余留的多边形递归地执行相同操作,直到所余留多边形为三角形。该算法执行简单、有效。在进行三角化的过程中,算法假设面片合并不会导致退化多边形[Hinker93]。然

而共线边等退化多边形实际上是简化过程中较为常见的现象。如果不对这些特殊情况进行特殊处理,会使算法的可靠性大大降低。[Kalvin96]考虑了退化多边形这种特殊现象,但它采用星型三角化的方法对超面重新剖分,因而不能处理含有孤岛的超面,而且会产生大量的狭长三角形。本章所提出的算法则采用了 Delaunay 方法,同时考虑了退化多边形以及带孔多边形等情况,适用范围广,且有很好的三角化效果(见本文 2.3.5)。

此外,与[Hinker93, Kalvin96]相比,本章所提出的算法的一个很重要的优越之处在于他们只能生成单一的简化模型,而我们的算法则基于层次分割的高斯球面可以在第一次生成的简化模型(基本简化模型)的基础上生成一系列粗化的模型(本文 2.3.6)。

2.2 高斯球面的层次分割

高斯球面是由网格模型上所有面片的单位法向映射到单位球面上而形成的。显然,高斯球面上法向的分布代表了网格模型的各面片的朝向的分布情况。由于本章提出算法的目的在于建立多层次细节模型,因此需要建立网格模型高斯球面的层次分割结构。

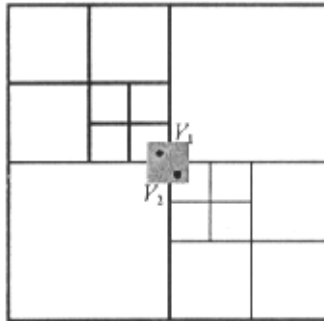


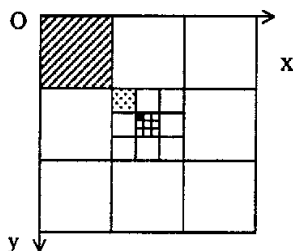
图 2.2 四叉树剖分

假设我们已经建立了高斯球面网格结构,则高斯球面上相邻网格的合并会驱动网格模型上近似共面区域的合并。因此,如果按照通常的四叉树剖分方法,球面网格的合并会导致生成不优化的近似共面区域。如图 2.2 所示为四叉树剖分的

网格，图中两点 V_1 、 V_2 分别表示两个相邻的三角形的法向在高斯球面上的映射。由图可知，只有最高层网格（所有网格合并为一个节点）同时包含这两个点，即直到网格合并的最高层，该两点所对应的相邻面片才会被合并，而实际上二者之间的夹角甚至小于最低层的网格误差。因此，我们需要一个可以实现八连通合并的层次结构，建立球面的层次网格。

在介绍我们的层次分割算法之前，作如下定义：

平面九叉树：平面九叉树表示一种层次数据结构。如图 2.3，正方形为树的根结点，其边长为 3^n 。将四边形等分为九个小正方形，其边长为原正方形边长的 $1/3$ ，即 3^{n-1} 。然后，将这九个小正方形分别递归分割，则得到平面九叉树。






注：模式 1: ；模式 2: ；模式 3: 

图 2.3 层次网格：平面九叉树(planar-ennetree)

我们的算法就是基于平面九叉树层次分割的思想对高斯球面进行剖分的。

首先，将一立方体映射到单位球面。然后对立方体各面分别进行平面九叉树剖分（图 2.3），并将其剖分结果映射到球面上，从而得到层次分割的球面。算法用三元组 (d_1, d_2, d_3) 存储球面网格编码。其中， d_1 从 0~5 取值，表示网格位于哪一立方体表面(如图 2.4)； d_2 、 d_3 表示网格在立方体表面上的位置（ d_2 对应 x 向、 d_3 对应 y 向），其取值范围依当前所选用的细节层次 $Level$ 而定，为 $0 \sim 3^{Level}$ 。这里，我们令 $MeshNum$ 代表一个立方体表面上沿一个坐标轴方向的网格总数，即 $MeshNum = 3^{Level}$ 。

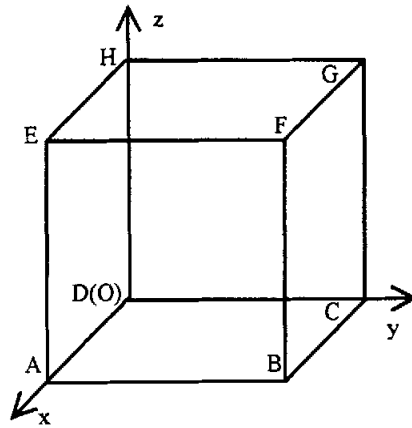
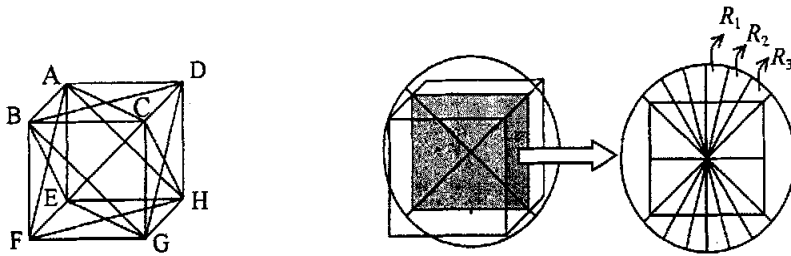


图 2.4 定义高斯球面网格编码 d_i : 位于面 ABFE 上的网格 d_i 值为 0; 位于面 CDHG 上的网格 d_i 值为 1; 位于面 ABFE 上的网格 d_i 值为 2; 位于面 ABFE 上的网格 d_i 值为 3; 位于面 ABFE 上的网格 d_i 值为 4; 位于面 ABFE 上的网格 d_i 值为 5;



(a) 分割球面的六个平面 ABGH, CDEF, AFGD, BCHE, BFHD, AEGC

(b) 利用大圆弧等弧度划分进行从平面到球面的映射, 其中 $R_1 = R_2 = R_3$

图 2.5 立方体表面网格到球面的映射

为了简化计算, 立方体到单位球面的映射定义如下: 如图 2.5(a)所示的 6 个平面与球面相交, 得到映射到球面上的立方体的 12 条棱边。这些边将球面分割成六个相等的曲面, 分别对应立方体的 6 个面。然后, 在球面上用大圆圆弧进行近似等立体角划分, 如图 2.5(b)所示。

图 2.3 所示为将要映射到球面上的立方体的一个表面。由立方体的六个表面形成的六个球面网格为层次球面网格最粗的一层 S^0 。以模式 1 标识的网格由 S^0 层

细分得到，属于 S^1 层；以模式 2 标识的网格由 S^1 层细分得到，属于 S^2 层；以模式 3 标识的网格由 S^2 层细分得到，属于 S^3 层… 由此整个球面网格形成一个以 S^0 层六个节点为根的森林结构。

我们采用层次分割的高斯球面来定义近似共面合并的误差阈值，该值即为高斯球面一个网格定义的最大法向夹角（度）： $\sqrt{2} \times 90 / 3^{Level}$ 。Level 值越小，网格越粗，误差阈值越大；Level 值越大，网格越细，则误差阈值越小。

2.3 基于高斯球的近似共面合并多层次细节模型

我们知道，高斯球面上法向的分布代表了网格模型各面片的朝向的分布情况。若两面片的法向在高斯球面上相距较近（小于给定阈值），则表明二者的朝向近似相同。若该两面片相邻，则二者位于一个近似共面区域内。正是基于这样的原理，本文提出了基于高斯球面的近似共面合并多分辨率模型。基于近似共面合并的模型简化算法虽然在保持模型几何特征、算法效率等方面有很多优越性，但这方面的研究很少，其中一个重要原因就是共面合并算法非常复杂，不易实现。本文通过建立高斯球面的层次分割结构，方便地实现了近似共面区域的合并和对模型的简化，进而生成了多分辨率模型。

在这里，我们沿用了Kalvin[Kalvin96]的超面的概念——如果相连通的若干面片中，任一对面片的法向夹角均小于用户定义的共面误差阈值，可被近似看作平面，并称之为超面。

算法以相邻面片法向夹角的大小作为判断共面的标准，基于高斯球面网格的层次分割，通过执行以下步骤生成多分辨率模型：

1. 根据网格模型中面片的法向把他们划分为一些面片簇，建立近似共面面片集合；
2. 在近似共面的面片簇内搜索超面及其边界；
3. 简化超面边界；
4. 对超面边界重新三角化，生成简化的模型作为基本简化模型；
5. 基于基本简化模型以及高斯球面网格层次结构，生成一系列较粗化的模

型，并与基本简化模型、原始模型一起成为多分辨率模型。

其中，算法通过分割高斯球面实现对网格模型中面片的划分，计算简单，容易实现；超面的重新三角化采用 Delaunay 方法，得到了较好的三角化效果；对位于超面边界上的邻近顶点，依据顶点的重要度对它们进行合并，可删除面积小、视觉重要度低的表面细节；采用八连通邻域面片合并的策略生成多层次细节模型，实现了较好的简化效果，同时，简化模型能很好地保持原模型的拓扑及几何特征。

我们将从建立共面面片集合、超面边界搜索、邻近边界点合并、重新三角化等部分对算法加以介绍。

2.3.1 建立近似共面面片集合

由于本算法首先生成简化程度最小的近似模型 $M^{MaxLevel-1}$ ，并以其为基本模型，利用高斯球面的八连通邻域合并生成多层次细节模型。因此，算法首先根据分割最精细的底层高斯球面网格 $S^{MaxLevel-1}$ ($6 \times MeshNum \times MeshNum$) 对原始模型中的面片分组。生成底层高斯球面网格，对大圆圆弧的等弧度分割角 $DeltaAngle = 90^\circ / (2 \times MeshNum)$ 。

设给定面片的法向为 $\vec{n}(x, y, z)$ ，则可确定该面片所对应的球面网格的编码 (d_1, d_2, d_3) (图 2.6)。算法首先根据法向三分量的大小判断面片的法向 \vec{n} 所在高斯球面网格的编码 d_1 的值，即确定 \vec{n} 在立方体的哪一表面，然后再在该面内，由 \vec{n} 对应的高斯球面上的点在坐标轴上的投影确定 d_2, d_3 。根据对应球面网格的编码，我们把各面片归类到相应的高斯球面网格中，形成面片簇。面片簇中的面片，可能相连通，或不连通。其中，相连通的面片组成近似共面集合。因此，一个面片簇与一个或多个近似共面面片集合对应。

```

Spherecode( $\vec{n}$ ) {
     $d_0 = WhichFace(\vec{n})$ ;
    switch( $d_0$ ) {
    case 0://x,z,y
         $Angle1 = \text{atan}(\vec{n}.z / \vec{n}.x)$ ;    $Angle2 = \text{atan}(\vec{n}.y / \vec{n}.x)$ ;
        break;
    case 1://-x,z,y  symmetrizing to +x
         $Angle1 = -\text{atan}(\vec{n}.z / \vec{n}.x)$ ;    $Angle2 = -\text{atan}(\vec{n}.y / \vec{n}.x)$ ;
        break;
    case 2://+y,x,z
         $Angle1 = \text{atan}(\vec{n}.x / \vec{n}.y)$ ;    $Angle2 = \text{atan}(\vec{n}.z / \vec{n}.y)$ ;
        break;
    case 3://-y,x,z  symmetrizing to +y
         $Angle1 = -\text{atan}(\vec{n}.x / \vec{n}.y)$ ;    $Angle2 = -\text{atan}(\vec{n}.z / \vec{n}.y)$ ;
        break;
    case 4://+z,y,x
         $Angle1 = \text{atan}(\vec{n}.y / \vec{n}.z)$ ;    $Angle2 = \text{atan}(\vec{n}.x / \vec{n}.z)$ ;
        break;
    case 5://-z,y,x symmetrizing to +z;
         $Angle1 = -\text{atan}(\vec{n}.y / \vec{n}.z)$ ;    $Angle2 = -\text{atan}(\vec{n}.x / \vec{n}.z)$ ;
        break;
    }
     $d_1 = MeshNum/2 + (\text{int})(Angle1/DeltaAngle + 0.5 * \text{sign}(Angle1))$ ;
     $d_2 = MeshNum/2 + (\text{int})(Angle2/DeltaAngle + 0.5 * \text{sign}(Angle2))$ ;
}

WhichFace( $\vec{n}$ ) {
    if (fabs( $\vec{n}.y$ ) ≤ fabs( $\vec{n}.x$ ) && fabs( $\vec{n}.z$ ) ≤ fabs( $\vec{n}.x$ )) {
        if ( $\vec{n}.x > 0$ ) return 0; //+x
        else if ( $\vec{n}.x < 0$ ) return 1; //-x }
    else if (fabs( $\vec{n}.x$ ) ≤ fabs( $\vec{n}.y$ ) && fabs( $\vec{n}.z$ ) ≤ fabs( $\vec{n}.y$ )) {
        if ( $\vec{n}.y > 0$ ) return 2; //+y
        else if ( $\vec{n}.y < 0$ ) return 3; //-y }
    else if (fabs( $\vec{n}.x$ ) ≤ fabs( $\vec{n}.z$ ) && fabs( $\vec{n}.y$ ) ≤ fabs( $\vec{n}.z$ )) {
        if ( $\vec{n}.z > 0$ ) return 4; //+z
        else if ( $\vec{n}.z < 0$ ) return 5; //-z }
    }
}

```

图 2.6 确定面片所属的高斯球面网格编码

2.3.2 搜索超面 (Superface) 边界

在对面片分类的同时, 算法还生成属于同一高斯球面网格内的景物网格模型的顶点簇。在搜索超面的边界之前, 先将各高斯球面网格中的顶点分类: 位于一个超面内部的顶点为内点, 存于高斯球面网格的内点表中; 属于两个以上超面的则为边界点, 存于边界点表中。然后, 在边界点表中搜索封闭的环, 并将属于环的点定义为该环的边界点。这里的环可能是一个超面的外边界, 也可能是超面的内孔的边界。内外边界的判断有很多方法, 本文算法采用的是先将环投影到最大投影平面上, 然后根据其走向 (顺时针或逆时针) 进行判断的方法。经过判断之后, 对内边界需找其匹配的外边界。与内边界匹配的外边界很容易由网格各点、面的连通关系找到, 在此不做赘述。

为了进一步删除冗余的顶点, 算法还采用了共面区域公共边界直线化、邻近边界点合并等原则, 简化超面边界。

2.3.3 超面边界直线化

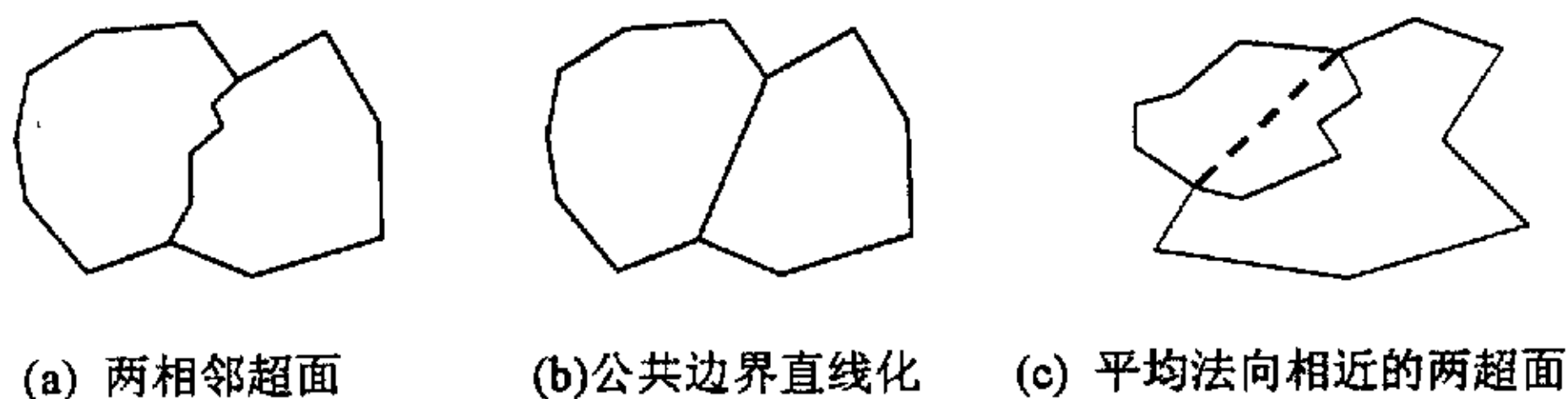


图 2.7 超面边界直线化

在允许的误差范围内, 我们可以认为超面是一个平面。注意到两平面交于一条直线, 因此两相邻超面的公共边界应近似为直线。根据这一原理, 我们对各超面的边界进一步简化, 如图 2.7 所示, 即用直线代替相邻超面的公共边界。

可以证明, 超面边界直线化产生的法向误差, 并不会超过超面所属细节层次的误差阈值。对于平均法向接近的两超面 (图 2.7(c)), 可能会有直线化的边界与原边界几何空间距离较大的情况, 但是, 在这种情况下, 直线化产生的几何空间

误差显然对模型的几何结构影响很小。实验结果表明,边界直线化可使网格模型得到大幅度简化。

2.3.4 邻近边界点合并

在经直线化的超面边界上,仍存在因点分布密度过大而产生的冗余现象。合并邻近边界点可以删除冗余的顶点,使模型得到进一步简化。然而,合并目标点的选取是顶点的合并操作普遍存在的问题。由于本算法的顶点合并只限于在空间上相距很近的点,因此可以直接选用两被合并顶点之一作为目标点。另一方面,由于多次合并可能导致较大的累积误差,若目标点选择不当,会引起模型整体形状的剧烈变化。

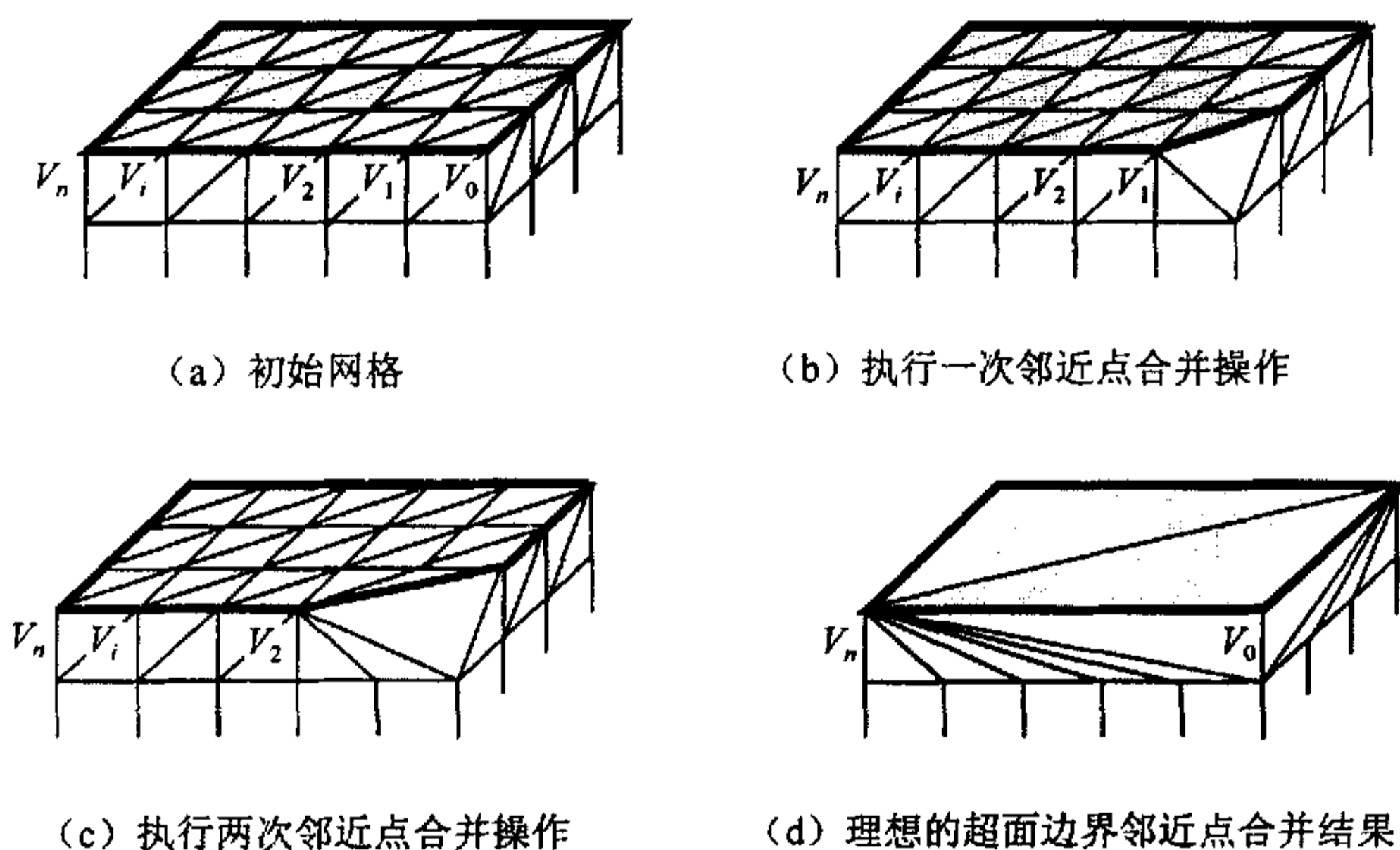


图 2.8 超面边界的邻近点合并
(注:有颜色标识的部分为超面;粗线标识的边组成超面边界。且设除标识部分外,其它部分不予特别处理。)

如图 2.8 (a) 所示为待简化的超面。由于超面边界相邻顶点 V_0 、 V_1 之间距离较小(假设小于所规定的阈值),则可执行合并简化。图 2.8 (b) 为执行很并简化后的情况。由图可见,网格模型的一个特征角被截除,使网格模型的形状发生

了一定的改变。由于点 V_0 、 V_1 之间的距离很小，则可以认为形状的变化并不明显。但是由于合并后的点 V_1 ，仍可能与其它点合并（图 2.8c）使形变扩大，最终导致特征完全消失。然而，我们进行超面边界的邻近点合并，希望产生如图 2.8d 所示的结果，即将边界顶点合并到特征顶点，在保持超面几何特征的前提下，保留尽可能少的顶点。

因此，我们需要对模型各顶点进行分类，以区分它们在定义模型几何特征、几何形状中的重要程度。

● 网格顶点的重要度分类

很多现有的模型简化算法，都涉及到顶点分类的问题，并且建立了多种不同的重要度定义方法。



图 2.9 Schroeder 的顶点重要度分类

Schroeder 根据网格模型的局部几何及拓扑性质把模型中的顶点分为 5 种类型：简单点 (Simple)、复杂点 (Complex)、边界点 (Boundary)、内边点 (Interior Edge) 以及角点 (Corner)（图 2.9），并由顶点的类型决定该顶点可否可被删除 [Schroeder92]。Isler 根据顶点周围三角形的法向差值，把顶点分类为平滑顶点 (Flat Vertices)、边点 (Edge Vertices)、特征点 (Feature Vertices) 三类，表征了顶点的三个视觉重要度层次 [Isler96]。他的基于三角形删除的多分辨率模型算法根据三角形三个顶点的视觉重要度 (Visual Importance) 决定三角形的删除顺序及删除操作的类型 (两个顶点合并或三个顶点合并)。Li 则定义了点合并的边删除、表面拉平、折线拉直三种变换操作，并取这三种变换操作所产生的误差的最小值为该点 P 的重要度 ($importance(P) = \min(Error(P)_I, Error(P)_{II}, Error(P)_{III})$) 用于决定模型简化中点删除的顺序 [Li98b]。这里的重要度实际上是简化误差的度量方

式。

由以上分析可见，不同的应用目的，需要不同的重要度定义方法。

由于在我们的超面边界邻近点合并算法中，可否进行合并以及合并目标点的确定需要顶点的视觉重要程度的信息。因此，我们采用与 Isler 类似的顶点重要度计算方法，根据超面边界顶点周围面片法向分布的情况将其分类为一般点、棱边内部点、角点（图 2.10）。

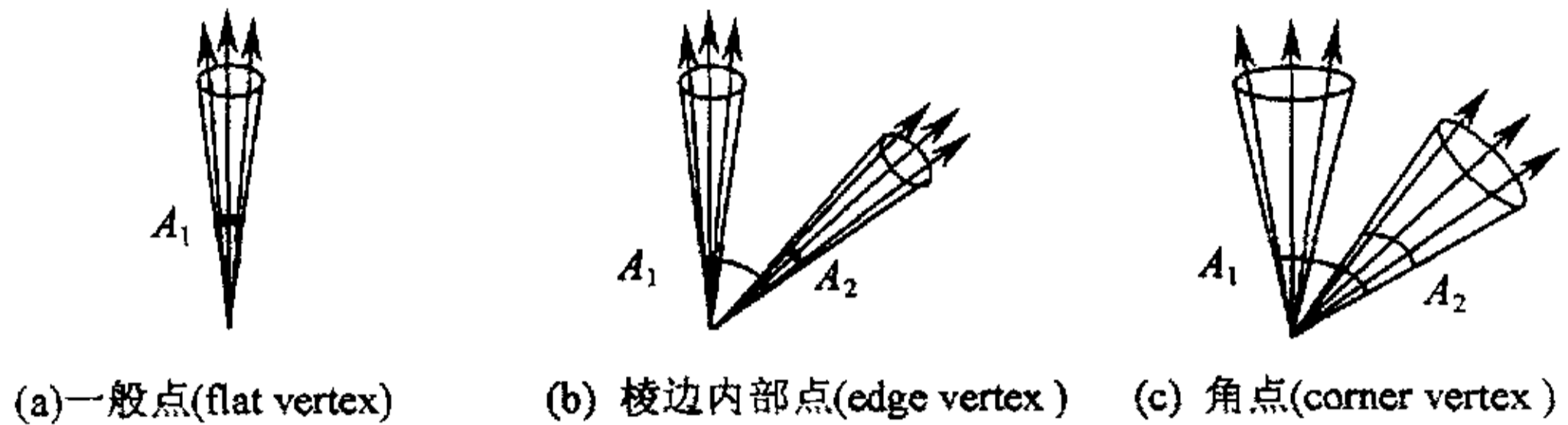


图 2.10 按顶点周围面片法向分布进行顶点分类

把顶点周围面片根据法向分组：首先找到法向夹角最大（ A_1 ）的两个面片，然后，以这两个面片分别作为两面片组的基准面片，根据与基准面片的夹角把其余面片分别归至两组中。两组内面片法向形成分别形成法向锥， A_2 取值为其中较大法向锥角度。

一般点：若 A_1 小于阈值，则该点为一般点（如图 2.10 (a)）。

角点：若 A_2 大于阈值，则该点为角点（如图 2.10 (c)）。

棱边内部点：若 A_1 大于阈值，且 A_2 小于阈值，则该点为棱边内部点（如图 2.10 (b)）。

其中，角点对定义模型的几何特征贡献最大，也就是说，角点的重要度最大，其次是棱边内部点，一般点的重要度则最低。

合并处理是在超面的边界环内进行的。在邻近边界点合并的过程中，将重要度低的点移动到重要度高的点。角点只能作为目标点，即只能将其它的点移到角

点，来实现边删除。若两点均为角点，则合并操作不执行。因为算法只对极近点合并，因此可以在实现简化的同时保持原景物模型几何、拓扑特征。

2.3.5 超面的重新三角化

给定一个区域的边界环，对该区域进行三角化有很多方法。本文算法采用的是先将边界环投影到最大投影面，然后在 2D 空间进行 Delaunay 三角化(图 2.11)。实验结果表明，采用该方法生成三角化网格，可尽量减少狭长三角形，三角化结果较好。

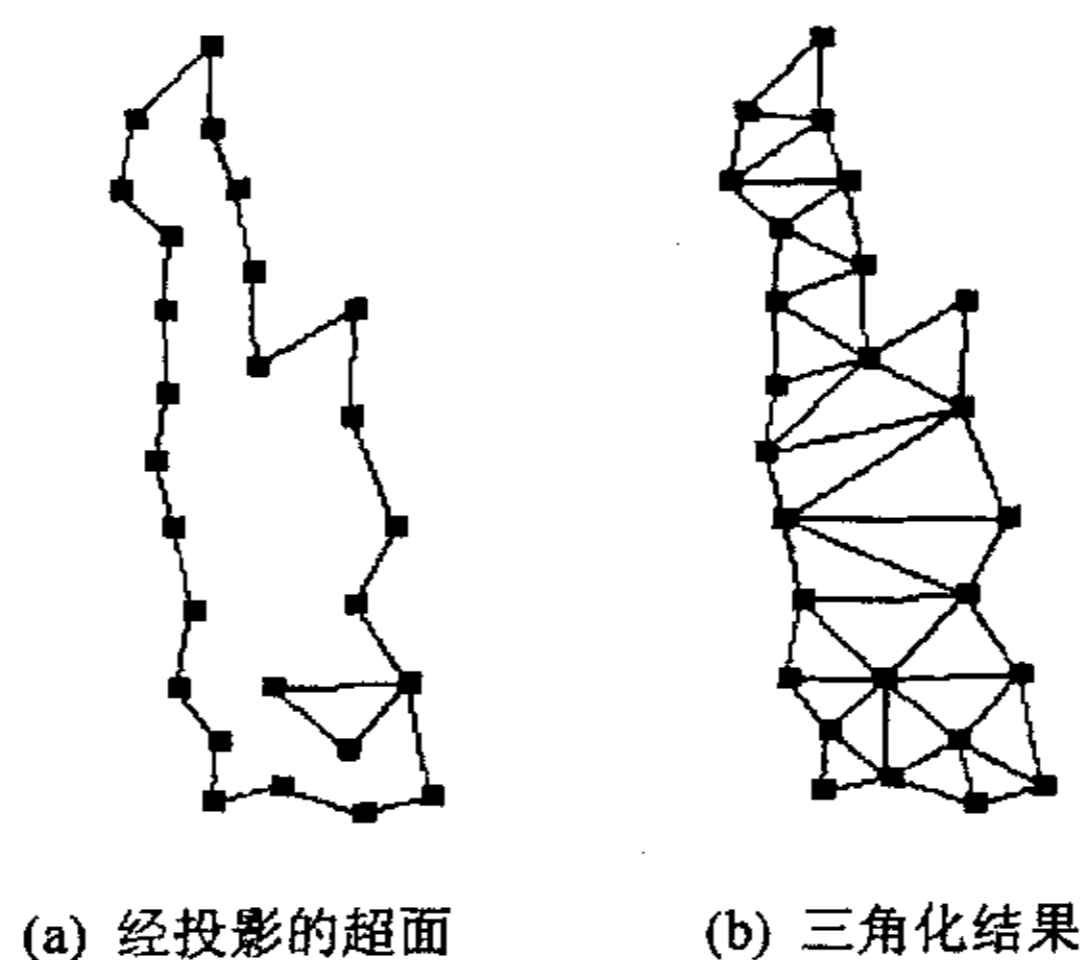


图 2.11 超面重新三角化

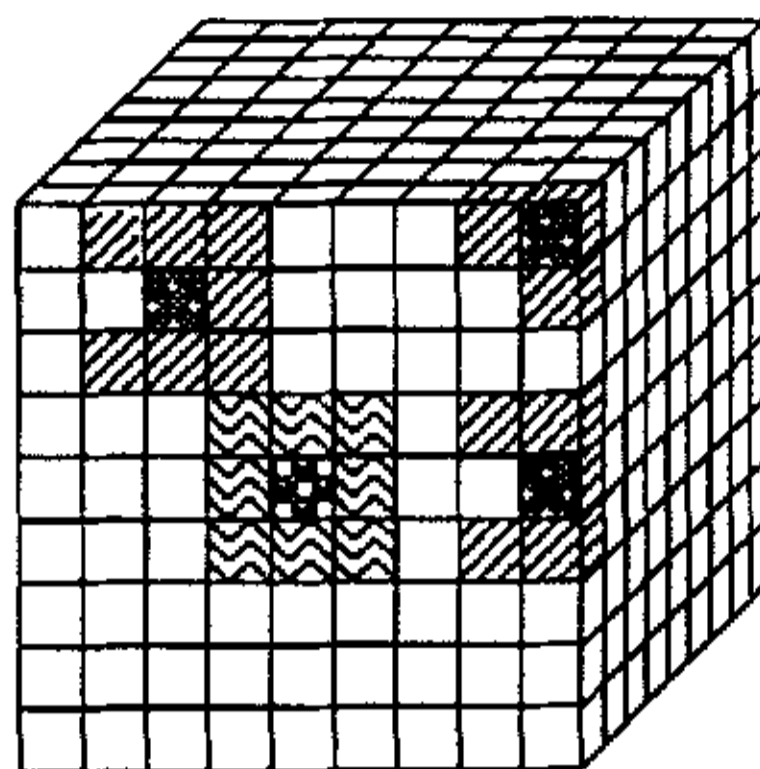
至此，已生成最精细的简化景物网格模型 $M^{MaxLevel-1}$ ($MaxLevel$ 是 LOD 的细节层次总数)。

2.3.6 多层次细节模型

由已知 M^i 层细节模型，生成 M^{i-1} 层(较粗)的过程，是通过合并高斯球面八连通邻域内各高斯球面网格中的超面实现的。

设图 2.12 所示为 S^i 层高斯球面网格所对应的立方体,其高层网格 S^{i-1} 的结点由 S^i 层结点合并得到。模式 4 及模式 5 标识的网格合并成的较粗网格为 S^{i-1} 层结点 $Mesh(S^{i-1})(i,j,k)$ ，那么 $Mesh(S^{i-1})(i,j,k)$ 的子结点 (S^i 层网格结点) 从左上结点开始

依次为 $\text{Mesh}(S^i)(i, 3*j, 3*k)$ 、 $\text{Mesh}(S^i)(i, 3*j, 3*k+1)$ 、 $\text{Mesh}(S^i)(i, 3*j, 3*k+2)$ 、 $\text{Mesh}(S^i)(i, 3*j+1, 3*k)$ 、 $\text{Mesh}(S^i)(i, 3*j+1, 3*k+1)$ 、 $\text{Mesh}(S^i)(i, 3*j+1, 3*k+2)$ 、 $\text{Mesh}(S^i)(i, 3*j+2, 3*k)$ 、 $\text{Mesh}(S^i)(i, 3*j+2, 3*k+1)$ 、 $\text{Mesh}(S^i)(i, 3*j+2, 3*k+2)$ 。其中， $\text{Mesh}(S^i)(i, 3*j+1, 3*k+1)$ ，即模式 4 标识的网格，为父结点 $\text{Mesh}(S^{i-1})(i, j, k)$ 的中心子结点；模式 5 标识的网格，为父结点 $\text{Mesh}(S^{i-1})(i, j, k)$ 的周边子结点。



注：模式 4：☒；模式 5：∞；模式 6：⊘；模式 7：∕

图 2.12 八连通网格结点合并

超面合并由以下几个步骤实现：

(1) 以中心子结点中的各超面为种子，向邻接超面扩展。若当前超面的邻接面（包括边邻接和点邻接）是其同父兄弟结点内的超面，则执行扩展，即当前的超面与其邻接超面合并，形成新的超面。这样的扩展递归地执行下去，直到当前超面不再有可合并的邻居为止。

如果高斯球面网格的各层，维持固定的边界，也就是说，景物网格模型的简化仅限于以中心子结点中的各超面为种子向邻近同胞兄弟结点扩展，产生的较粗景物网格模型的共面集合可能并不合理。例如，即使两个面片的法向十分接近，如果它们分别对应不同立方体表面（ d_i 不同），直到简化的最高层 M^0 层，两面片也不能被合并。为解决这种情况，算法除以中心子结点中各超面为种子扩展外，还以周边子结点中各超面为种子进行扩展。

(2) 以周边子结点中的各超面为种子，向邻接超面扩展。如图 7 中，模式 6

标识的网格为父结点的周边子结点，除网格自身内面片外，它们可合并周边网格中以模式 7 标识的网格结点，即除其父结点的中心子结点外的八连通邻域内的网格结点内的超面。这样的扩展递归地执行下去，直到当前超面不再有可合并的邻接超面为止。

以上两步骤中，扩展涉及的超面与扩展最终生成的超面为父子关系。

(3) 把 M^i 层结点中的超面转存到各自的父结点，即 M^{i-1} 层结点中。然后，对各超面进行边界取直及三角化等操作，最后生成满足 M^{i-1} 层阈值的 M^{i-1} 层简化网格。

(4) 递归生成其它层次简化网格，直到最粗层 M^0 层。

2.4 实验结果

该算法已在 Pentium 350 上用 C 语言实现，并成功地应用于不同的模型，图 2.13、图 2.14 分别是算法对恐龙模型和牛模型简化的结果，所取 $MaxLevel = 5$ ，图中 (a)、(b)、(c)、(d)、(e)、(f) 依次为原模型及 M^4 、 M^3 、 M^2 、 M^1 、 M^0 五个层次细节模型。由图可见，即使最简化的 M^0 层细节模型，也能很好地保持原模型的几何特征。应用该算法建立恐龙模型、牛模型层次结构所用的时间分别为 190.89040 秒和 8.63359 秒。

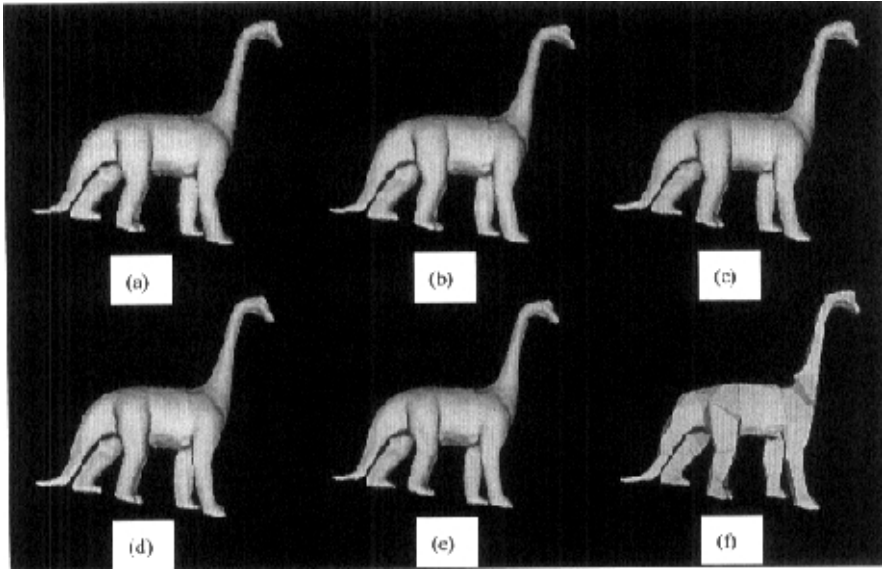
2.5 小结

本章讨论了一种采用近似共面合并的多层次细节模型生成算法。算法基于法向近似面片合并原则，可在层次分割高斯球面的基础上生成多层次细节模型。算法在预处理阶段执行。

在视点变换时，可根据当前的视点参数，动态生成包含不同细节的层次模型。这是由于算法严格定义了各层次超面之间的父子关系，在实时显示过程中，可根据当前视线方向，对高斯球面的不同部分，选用不同的细节层次，从而生成在该时刻保持视觉等价的包含不同细节的场景模型。设光线方向为 L ，对于高斯球面上法向 N 与 L 夹角 θ 很小的区域，我们可以取分割较粗的网格单元，即以较大的

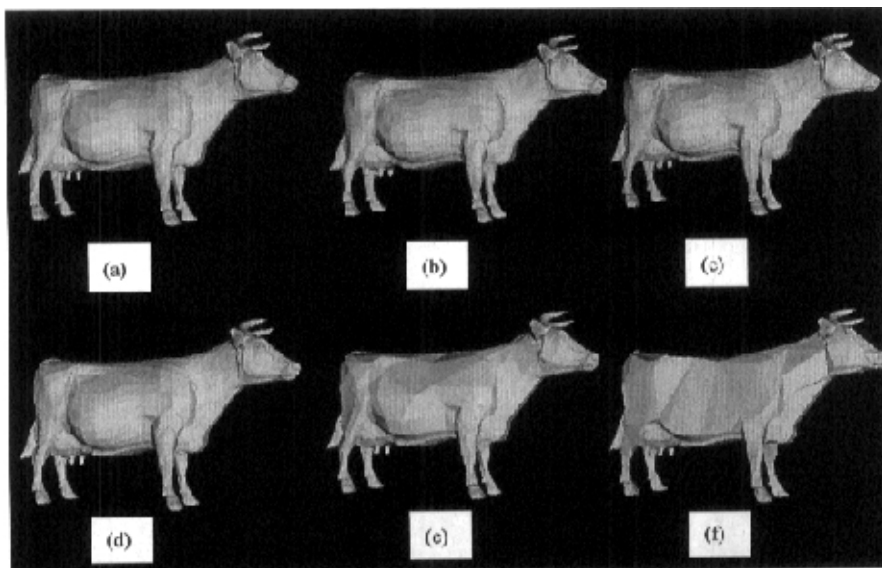
阈值对网格模型上的面片进行合并。这是由于当入射角 θ 很小（接近 0 度）时，余弦函数 $\cos(\theta)$ 对 θ 的变化不敏感， θ 的变化对所绘制模型的视觉效果影响较小。反之，若入射角接近于直角，余弦函数随角度变化明显，因而对取较大入射角的高斯球面区域，需要采用较细的网格单元，即以较小的阈值对网格模型上的面片进行合并。

至于如何根据景物面片与视点的距离及方位选取恰当的阈值以控制不同景物面片细节显示的层次，则待继续做深入研究。



- | | |
|--------------------------|--------------------------|
| (a) 原数据集: 47, 858 个三角形 | (b) M^4 : 47, 776 个三角形 |
| (c) M^3 : 46, 538 个三角形 | (d) M^2 : 32, 100 个三角形 |
| (e) M^1 : 7, 690 个三角形 | (f) M^0 : 3, 447 个三角形 |

图 2.13 算法对恐龙的简化



- | | |
|-------------------------|-------------------------|
| (a) 原数据集: 6, 208 个三角形 | (b) M^4 : 6, 038 个三角形 |
| (c) M^3 : 6, 002 个三角形 | (d) M^2 : 5, 706 个三角形 |
| (e) M^1 : 3, 750 个三角形 | (f) M^0 : 2, 500 个三角形 |

图 2.14 算法对牛的简化

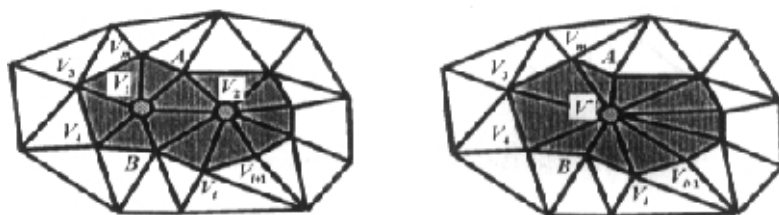
第三章 基于分割插值的连续多分辨率模型

3.1 增量式多分辨率模型

本章所研究实现的“基于分割插值的连续多分辨率模型”是一个粗化算法 (Decimation Algorithm)。算法以原始模型为初始值, 自动执行边删除操作, 递归地删除其中的顶点及面片。这种方法也称作增量式多分辨率模型。

边删除操作对网格模型的修改如下:

1. 把边两端点移至新点 \bar{V} ;
2. 把与边任一点相关的边、三角片的对应顶点用新点代替;
3. 删除褪化的边及三角片。



(a). 边 V_1V_2 的局部相关区域

(b). 边 V_1V_2 的简化结果

图 3.1 边 V_1V_2 的删除

经过以上三个步骤, 完成对网格模型的几何及连结关系的修改。如图 3.1 所示, 边 (V_1, V_2) 的删除变换, 即将其两端点 V_1, V_2 合并至新点 \bar{V} , 删除 V_1, V_2 ; 把三角形 $\Delta V_i V_1 V_j$ 和三角形 $\Delta V_i V_2 V_j (i \neq j, \text{ 且 } i, j \in \{3, 4, \dots, m\})$ 的顶点 V_1, V_2 分别改为 \bar{V} ; 边 (V_1, V_2) 褪化为一点 \bar{V} , 而包含该边的三角形 ($\text{Container}(V_1V_2)$) $\Delta B V_1 V_2, \Delta V_2 V_1 A$ 。褪化为边, 将其删除。由图中我们可以看出, 边删除是对网格模型的局部操作, 并只对模型的局部区域产生影响。执行一次操作, 模型中顶点数目减少一个、三

角形的数目减少一到两个（删除边界边，减少一个三角形；删除内部边减少两个三角形）[Ronfard96a]。

一般的，几何元素的删除操作不会引起模型拓扑结构的改变。Garland[Garland97]和 Popović [Popović97]在 1997 年分别提出的基于边删除的简化算法的扩展算法则把一般的边删除操作扩展到点对（Vertex Pair）的删除操作（图 1.7）。即对没有边相连的两个顶点，也可以执行合并操作。这样，基于点对的删除操作可以连通不同的拓扑结构、封闭模型表面的空洞等，实现对模型拓扑结构的简化。显然拓扑简化可以在一定程度上提高模型的简化程度。但在有些领域如医学可视化等，在对模型进行简化时，保持细小的拓扑结构如小洞的重要性甚至高于对模型整体几何形状的保持。本章所提出的多分辨率模型是基于狭义的边删除操作，因此不会进行拓扑简化。但正如本文在 1.3 节所述，该算法可以扩充到点对的删除操作，从而实现拓扑简化。

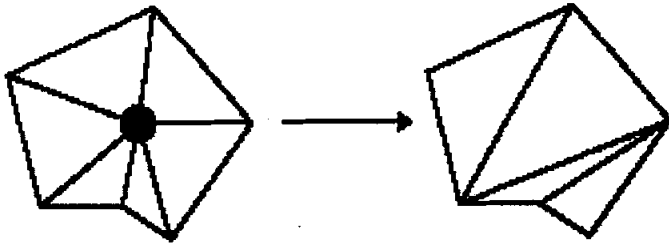


图 3.2 顶点的删除 (Vertex Contraction)

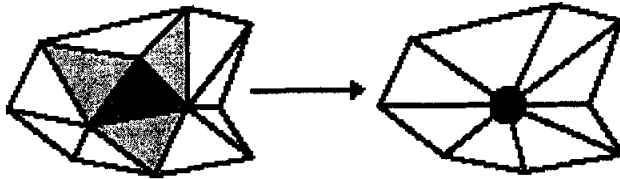


图 3.3 三角形的删除(Face Contraction)

实际上，除了边删除算法，顶点及三角形删除算法也都是利用递归删除算法生成增量式多分辨率模型。如 Schroeder 提出的基于顶点删除的网格简化方法 [Schroeder92]也是典型的增量式算法（如图 3.2）。首先根据点的局部拓扑、几何

信息将各顶点分类（简单点、复杂点、边界点、内部点、角点），然后删除满足删除标准（大于点到平均平面，或点到边的距离阈值）的顶点，并对由此产生的空洞进行三角剖分。[Zhou96, Li98]也在基于点删除的网格模型简化方面做了工作。Hamman 等则提出了基于三角形删除的多分辨率模型[Hamann94]，通过将三角形的三个顶点合并为一点，并删除退化的相邻三角形和原三角形对三角网格模型进行简化[Hamman94, Ma98, Zhou98]（如图 3.3）。

在对网格模型的简化过程中，几何元素（点、边、面片）的删除操作是对网格的局部简化，因此只对局部区域内的网格连结关系和几何结构等产生影响。在介绍本论文所提出的基于边的简化算法之前，我们有必要给边的局部相关区域加以定义。

边的局部相关区域：边的局部相关区域定义为其两端点相邻三角面片的并集。如图 3.1 所示，边 V_1V_2 的局部相关区域定义为点 V_1 、 V_2 相邻三角面片的并集 $Neighbour(V_1V_2)$ ，即由阴影标识的三角面片。相关区域内的网格顶点即这些面片的顶点，如图为 V_1 、 V_2 、 V_3 、 V_4 、…… V_i 、 V_{i+1} 、…… V_m 。其中 V_3 、 V_4 、…… V_m 定义为边 V_1V_2 的相邻点。

与其它增量式简化算法类似，我们所提出的算法通过递归地执行简单的几何元素（边）删除操作实现对整体模型的简化。算法步骤如下：

1. 判断各边是否可删除，建立待删除边集合。
2. 计算待删除边集合内各元素对应新点的位置，并求取该边的简化误差 ($cost$)。
3. 根据简化误差，对待删除边集合中的各边排序，建立队列。误差值最小的边列于队首。
4. 从待删除边队列中取出位于队首的边。将其从网格中删除，用在 2 中计算所得的新点代替该边。
5. 重新计算相关边的对应的目标点以及简化误差，调整其在队列中的位置。
6. 递归执行 4、5，直到简化至用户定义的面片数，或达到给定的简化误差阈值。

由以上的算法执行步骤可知,该简化算法是贪婪算法,在执行一个简化操作之后便不再考虑该简化操作所删除的几何元素。通常,贪婪算法不会产生最优化的简化结果。实际上,求曲面的优化近似模型是一个NP-Hard[Agarwal94,Cormen90]问题,到目前为止还不存在可生成最优简化模型的算法。采用更加细致的搜索方案也许可以生成比贪婪简化模型更好的结果,但也不能达到最优化的程度。

显而易见,递归简化算法使我们可以建立原始模型的顶点与各中间简化模型的顶点之间的对应,而这一对应关系完全决定了从原始模型到任意简化模型之间的变换,即决定了从原始模型到最终简化模型之间的映射。近似模型上的每一顶点以及相邻关系都是与原始模型上的顶点、面片相对应的。这样,递归简化算法自然而然地生成了模型的层次结构,即多分辨率表示。

在增量式多分辨率模型算法中,模型中几何元素的简化顺序及简化后点的分布是影响简化效果的非常重要的两个方面。在现有的算法中,简化后的模型顶点通常为原模型顶点的子集。简化过程中生成的新点,则简单地采用位于原始模型边上的点或沿用原模型的顶点。Turk 对新点的位置进行了一定的优化[Turk92],根据模型表面的曲率确定新点的位置,取二次误差最小的点或简化边上的点为新点。采用以上方法定义新点的位置,虽然能在一定程度上保持原模型的形状特征,但不可避免地会导致模型的单调收缩,从而产生越来越大的 Hausdorff 距离。

对于几何元素的简化顺序,现存的算法通常采用统一的简化误差计算策略进行控制。我们认为,这种一贯制的简化控制方法并不符合简化过程的需要。在增量式多分辨率模型的计算过程中,在不同简化阶段,需要侧重不同的方面。本文针对以上两个方面的问题,提出了加权和函数控制的基于插值的增量式多分辨率模型,有效地对多边形网格模型进行了简化处理,取得了良好的效果。3.3节、3.4节将分别对算法所采用的边简化顺序的控制原则以及新点位置的确定方法进行详细介绍。

3.2 建立待删除边集合

3.1 节中提到增量式多分辨率模型的简化算法的第一步就是建立待删除边集合。一般的，模型中所有边都可作为简化操作的对象。对于不存在较多细节且表面平滑封闭的网格模型，把所有边列为待简化对象，在其中按照算法误差判定方法查找较优的边，并对它进行删除操作，不会对简化结果产生很大影响。但是，对于存在较多表面细节或存在边界的网格模型，简单地以单一的误差判定方法保持网格模型的几何特征并不能达到理想的效果，甚至会产生不合法的简化。

如图 3.4a 所示，由若干三角面片组成并在中心位置存在小孔的正方形表面，边(V_0, V_1)的长度在整个网格中最小，若根据边长最小原则该边应最优先删除。但是将该边折叠为一点会生成非流形的网格结构（如图 3.4b），因此应避免执行对该边的删除。同理，对于存在较多几何特征的模型也需要采取措施以防止删除对保持拓扑结构及几何特征较为重要的边。

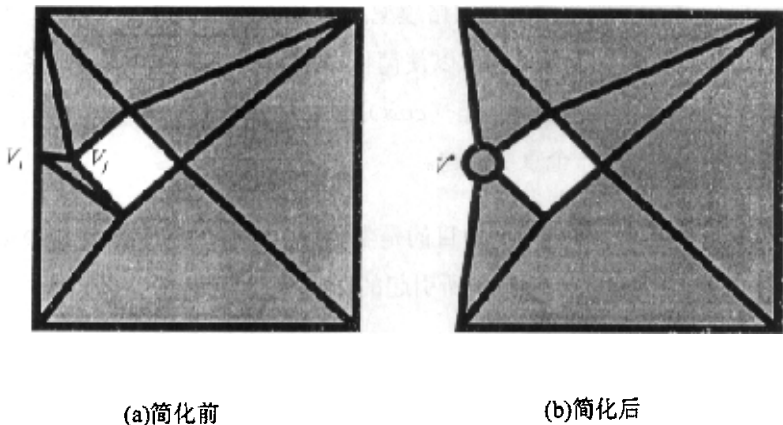


图 3.4 产生非流形网格结构的边删除操作

因此，本算法首先定义各顶点的重要度，然后根据两端点的重要度判断边是否可以被删除。

我们根据网格顶点所在局部区域的曲率大小，即周围面片法向分布的情况，将其分类为一般点、棱边内部点、角点（本文 2.3.4 节），并定义相应的重要度。其中，一般点重要度设为 0；角点重要度设为 2；棱边内部点重要度设为 1。若由

以上定义得到一点重要度为 $M (M \neq 2)$ ，且为模型边界上的点，则该点重要度为 $M+1$ 。若一边两端点重要度均大于等于 2，则不能被删除；否则，将其加入待删除边集合。

在本算法中，边端点重要度同时会影响简化生成新点的位置。对此，本文将在 3.4 节讨论。

3.3 简化误差的度量及控制函数

众所周知，我们进行模型简化的目的是生成与原始模型 M 之间的几何误差不超过给定阈值 ϵ 的最简化的模型 M^* ，即

$$\|M - M^*\|_{\infty} \leq \epsilon。$$

正如经典的 Knapsack 问题 [Sedgewich83]，我们的目标函数 (Objective Function) 为尽可能多地删除模型中的顶点以生成尽可能简化的模型，能力函数 (Capacity Function) 则为维持原网格模型几何形状的容差能力。显然，我们应该首先执行能力消耗较小的简化操作以使简化模型的误差以最慢速度增长。因此，如何度量各简化操作所引起的误差 (cost) 进而控制几何元素的简化顺序是建立增量式多分辨率模型的一个重要问题。

由于我们建立多分辨率模型的目的是生成与原始模型近似的简化模型，简化操作所产生的误差应该反映该操作所引起的网格模型的变化。

很多应用于绘制的简化算法，并不以景物空间的几何误差来度量简化误差，而是以屏幕空间的误差作为简化操作的误差 [Hoppe97, Luebke97]，建立视点依赖 (View-Dependent) 的简化模型。但是，由于基于形状的误差度量不仅便于计算而且很适合于有限元分析等非绘制方面的应用，因此大多数模型简化的研究人员都采用几何误差度量方法。而 Hausdorff 距离是目前常用的一种评价几何误差的方法 (本文 1.4 节)。一般来说，它也可以应用于简化算法的计算过程，对各简化操作进行评价，以寻找优化的简化顺序。但是由于精确计算的 Hausdorff 距离具有很高的算法复杂度，并且在迭代算法的每一步骤都要执行 Hausdorff 距离的计算，因而在迭代的简化算法中把 Hausdorff 距离作为简化操作的评价标准会使算法的运算速

度大幅度降低。

Hoppe等提出的基于能量优化的算法把面片数量、模型几何偏移误差以及启发式弹性能量之和作为能量函数。其中的几何偏移误差 $E_{dist}(K, V)$ 即以原始模型到简化模型的单向离散Hausdorff距离 $E_{dist}(M, V) = \sum_{v \in V} d_v^2(M)$ (M 为简化模型; V 为原始模型的顶点集合。)来度量[Hoppe93, Hoppe96]。由于弹性能量同时限制了简化模型相对于原始模型的几何偏移, 因此采用Hoppe的方法可以获得比较优化的简化结果。但是由于Hausdorff距离的计算复杂, 即使计算离散的局部Hausdorff距离在时间上的开销也很大, 因此Hoppe的算法运算速度很慢。为了进一步简化计算, [Soucy96a, Klein96, Ciampalini97, Kobbelt98]等定义了局部区域内计算点到模型距离 $d_v(M)$ 的方法。在实际应用中, 根据具体情况把距点 V 最近点的查找范围限制在模型 M 上一个小的局部区域 R , 然后计算 $d_v(R)$ 。而很多简化算法在简化过程中都会建立近似模型上的相邻顶点与原始模型上顶点的对应关系, 这样就可以很方便地计算到局部区域的最小距离, 即局部Hausdorff误差。局部区域内的Hausdorff误差, 在一定程度上提高了运算速度, 但同时简化模型的质量也有所下降。

Rossignac[Rossignac93]、Luebke[Luebke97]等利用模型空间的单元剖分对简化误差进行控制。单元剖分把模型的顶点分为不同的组, 当把一组顶点聚合为一点, 所产生的最大误差是聚合目标点与该组顶点之间的最大距离。显然该值不大于所在空间单元的对角线的长度。因此可以通过控制简化单元的尺寸了保证简化模型与原始模型之间的Hausdorff距离。Cohen等提出的Simplification Envelopes的方法则通过建立误差带, 把简化模型顶点偏移的范围限制在误差带内来控制简化模型与原始模型之间的Hausdorff距离[Cohen96]。

另一种常见的加速误差估计的算法通过生成局部近似曲面, 并用到近似曲面的距离来代替到网格模型的距离。如Schroeder等用点的相邻三角形的法向 \vec{n}_i 、中心点 \vec{x}_i 以及面积 A_i 来定义平均平面 (Average Plane)

$$\vec{N} = \frac{\sum \vec{n}_i A_i}{\sum A_i}, \quad \vec{n} = \frac{\vec{N}}{|\vec{N}|}, \quad \vec{x} = \frac{\sum \vec{x}_i A_i}{\sum A_i},$$

并用点到平均平面的距离 (Distance to Average Plane) 来判断点的简化误差

[Schroeder92]。此外，Gieng等还采用了高阶的拟合面片近似代替网格模型局部区域[Gieng98]。但是，点到近似曲面的距离计算的时间开销会随着阶数的提高而大幅度提高，并且对局部区域的高阶曲面拟合也较难实现。因此人们常把局部网格模型投影到平面，并把该局部模型转为高度场模型，然后用最小二乘法生成二次拟合曲面[Gieng98, Lansaster86]。

前面所述的简化算法都是以简化操作引起的模型顶点移动的距离作为简化误差，计算简单，可以显著提高运行速度，但是都相当保守。假设有一被密集网格化的平面模型，尽管在简化过程中网格模型的顶点会在该平面上产生移动，但整个模型还是保持为一个平面，因此简化模型与原始模型之间完全相同。而计算所得的Hausdorff误差却可能为一较大值。Ronfard提出把顶点沿垂直于相邻三角形方向的移动距离，即新点到简化操作所涉及的局部区域内所有面片的最大距离作为简化误差[Ronfard96a]。Garland则把新点到面片的最大距离改为距离的平方和，从而使计算更加简便，易于在迭代过程中实现对误差的控制[Garland97]，实现了计算时间与简化效果之间的协调统一。但是，Garland的算法中的二次误差只考虑了新点到相邻面片的距离平方和，而忽略了简化边 e 的长度、三角形的形状等因素，这样会导致局部区域的过度简化，以及狭长三角形的产生。

此外，很多简化算法还抛开了常用的几何误差的概念而采用边长[Xia96]、二面角[Algorri96]、局部曲率[Hinker93]等其它更简单的方式计算启发式误差来控制简化顺序。这些误差度量方案基于“短边的视觉重要度低”的思想，形式简单、计算方便，在很大程度上提高了算法的计算速度。但这类误差度量方法不能全面反映简化操作所引起网格模型改变的情况。Kobbelt利用显式定义的平滑函数（Fairness Function）来估计简化误差，优化简化操作的顺序[Kobbelt98]。他把反映模型的局部平滑程度的三角形形状系数 $R(P)$ （Roundness，内圆半径与最长边的比值）、反映局部曲率的相邻三角形所夹的二面角 $S(P)$ （Dihedral Angle）以及局部近似误差 $E(P)$ 组合为一个加权函数

$$F(S) = \sum_{P \in S} \alpha E(P) + \beta R(P) + \gamma S(P)。$$

其中， S 为简化模型， P 为模型中顶点。Kobbelt认为，对于不同的应用领域可以选用不同的权值，以实现保持尽可能多的表面细节、较好的三角形形状以及平滑的

模型表面等不同的目的。

- 可调加权控制函数

我们研究的主要目的在于快速生成高质量的简化模型，因此需要计算简单的误差度量机制有效估计简化操作所引起的误差以确定模型简化的顺序，进而生成高质量的简化模型。在采用增量式多分辨率模型时，在不同简化阶段需要不同的简化原则，以实现在简化效果以及计算速度两方面的统一。

在前面所分析的简化算法对简化误差的估计策略中，采用Hausdorff距离等精确的几何误差度量方法可以得到较好的简化结果，但是，运算时间很长。而基于局部Hausdorff距离度量误差的算法或其它简单的启发式误差度量方法虽然在运算速度上有所改善，但不能全面反映当前近似模型与原始模型之间的误差，因此会影响简化模型的质量。并且，现有的所有算法在整个简化过程中，都是采用统一的简化误差计算策略对几何元素的简化顺序进行控制。

与Kobbelt的方法类似，我们以反映模型不同局部几何性质的参数的加权平均作为控制函数，指导模型的简化顺序，使控制函数值小的边得到优先删除。其中，我们沿用了Kobbelt的反映模型局部曲率和局部变形程度的二面角误差和三角形形状系数误差两项，并用简化边的边长以及二面角误差两项来对模型的局部近似误差进行控制。由于简化边的边长计算简单，使算法的效率得到了提高。这样，控制函数 $F(e)$ 即为简化前后三角形转动的二面角 $D(t_i)$ ($t_i \in Neighbour(V)$)、简化边的长度 $L(e)$ 、简化边邻接三角形的形状系数 $S(e)$ 等三个方面加权平均：

$$F(e) = C_d \max_{t_i \in Neighbour(v)} (D(t_i)) + C_s S(e) + C_l L(e)$$

其中 V 表示边 e 简化后生成的新点； $S(e)$ 则为简化边 e 的邻接三角形的最大形状系数：

$$S(e) = \max_{t_i \in Container(e)} Roundness(t_i)$$

我们将边 e 的邻接三角形的形状系数定义为边 e 与另外两边中较长者 (L_{max}) 之间的长度之比 (L_e / L_{max})。

在这里，用二面角误差可以使模型的特征边特征点得到很好地保持； $S(e)$ 使狭长三角形优先删除； $L(e)$ 则把边长引为简化操作的依据。 C_d 、 C_s 、 C_l 三个权值

可根据模型的具体情况以及简化结果的要求定义。在模型的初始简化阶段，我们可以通过定义较大的 C_d 来，使较平坦区域的边首先得到简化，保持模型的形状特征，甚至小尺寸细节；当模型简化到一定程度，尽管相邻面片二面角允差已达到较大值，但如果 C_d 维持较大值，面片法向变化剧烈区域的小尺寸细节仍被保持，从而导致细节分布不均匀现象。算法这时对三个分量的权值进行调整， C_d 的值降低，同时提高 C_s 、 C_l 的取值，用经调整的控制函数重新计算各待简化边的控制误差，并依该值对简化队列中的边重新排序。新的简化队列使较小尺寸细节得以优先简化，而保持模型整体形状，实现远视距视觉等价的简化效果。

此外，为避免局部区域的过度简化，我们采用设定独立集的方法。即对一条边 e 进行简化，生成新点 V ，冻结 $Neighbour(V)$ 中各三角形的边，直到简化队列中所剩边均被冻结，将这些边同时解冻，继续进行简化操作。

3.4 新点位置的计算

在基于边折叠的多分辨率模型简化算法中，目标点的选取是另一重要的问题。目前大多数简化算法都选用简化边的一个端点作为目标顶点并保持原端点的坐标，这样任一简化模型的顶点集合都为原始模型顶点的子集。显然，我们可以通过优化目标点的位置获得较好的近似模型。Garland把二次误差最小的点作为目标点，使简化模型上新顶点的位置在一定意义上得到了优化[Garland97]。

采用原始模型顶点的子集还是优化的目标点位置，不同的应用领域有不同的要求。如果需要生成渐进网格模型[Hoppe96]，由于前一种方法不会引入新的顶点，不需要附加存储新点的位置，从而节省了大量的存储空间，具有一定的优越性。采用优化目标点则可使简化模型在整体形状上更接近于原始模型，并具有更好的三角形形状（更接近等边），且各面片面积趋于相等。因此，该方法适于生成固定的简化模型。

本文采用插值分割的方法求取新点的位置。在介绍我们的方法之前，先对分割方法进行简要介绍。

3.4.1 任意拓扑网格模型的分割

利用分割 (Subdivision) 的方法进行曲面造型是近年来十分活跃的一个研究领域。这种分割的思想最早起源于四十年代末、五十年代初 G. de Rham 用于描述曲线的“割角法” (Corner Cutting)。后来, Chaikin 提出通过添加新点和边不断细化多边形, 生成曲线 [Chaikin74]。1978 年, Catmull [Catmull78] 和 Doo [Doo78] 把这一思想推广到曲面, 标志着基于分割进行曲面造型研究的开始。分割曲面在计算机图形学、计算机辅助几何设计 (CAGD) 等方面有着广泛的应用, 并与多分辨率模型以及小波 (Wavelets) 等传统的数学工具等都有非常密切的联系。

分割算法通过递归地添加顶点、边、面片细化初始控制网格, 使控制网格收敛到曲面。其中, 添加的新点是由模型上临近顶点的线性组合而得到的。根据不同的添加点、边、面片的规则, 其极限曲面可以具有连续、 C^1 连续, 甚至 C^* 阶连续等优良的性质。到目前为止, 人们已经提出了很多种分割算法的分割策略, 可以根据细化准则 (插入顶点或割角)、生成网格类型 (三角形或四边形) 以及插入新点的方法 (近似或插值) 对它们进行分类。

当较粗网格的顶点同时也是细化网格的顶点时, 该较粗网格上每一个顶点都有一系列不同分割层次模型上的点与之对应。若对应于一个顶点的一系列点全部相同, 则分割算法用插值 (Interpolating) 的方法插入顶点。否则, 以近似 (Approximation) 的方法插入顶点 [Zorin99]。由于插值分割方法中, 定义初始控制模型的顶点全部都保留在极限模型上, 极有利于对造型的控制, 并且算法非常简单, 因此引起研究人员的极大兴趣。另一方面, 近似分割法生成的曲面模型质量较高, 且可以较快地收敛于极限模型, 与分割插值相比在这两个方面存在一定的优势。

规则顶点和特殊顶点对于分割算法是非常重要的概念。在这里, 我们首先阐述规则顶点及特殊顶点的定义 [Zorin99]: 对于三角形网格模型, 分割算法的规则分割策略生成的新点若为内点则价数为 6, 若为边界点则价数为 4。同理, 对于四边形网格模型, 顶点插入法和割角法生成的新点若为内点则价数为 4, 若为边界点则价数为 3。这样, 经过若干次分割, 网格中的大部分顶点的价数都属于以

下四种情况中的一种：三角网格上的 6 价内点或 4 价边界点、四边形网格上的 4 价内点或 3 价边界点。有以上四种价数的顶点称为规则点 (Regular)，否则称为特殊点 (Extraordinary)。

一般的，任意拓扑网格的分割算法是基于规则分割策略 (Regular Subdivision Scheme)。对于特殊的顶点或面片，需要定义附加的规则。正是这些分割策略和附加规则使得分割的极限模型有不同的光滑程度。

- 近似分割策略 (Approximation Subdivision Schemes)

由 Doo[Doo78]和 Catmull[Catmull78]提出的最早的分割算法都基于四边形网格，并分别是对二次和三次 B 样条的推广。Doo 用分割过程的傅立叶变换 (Fourier Transforms) 和特征分析 (Eigen analysis) 对特殊顶点附近的网格的几何性质进行了分析 [Doo78]。后来，Ball 推导了一些必要条件，把这一方法进行了扩展 [Ball88]。此外，Reif 也对特殊点附近的网格进行了分析，并推导出了 C^1 连续的充要条件 [Reif95]。Loop 则基于二次三角 B 样条的推广首先提出了任意三角网格的分割算法 [Loop87]。还有一些任意拓扑网格的分割算法试图通过一些割角算法规范拓扑或限制网格的连结关系来直接生成在全局上达到若干阶连续的一系列样条面片 [Loop94, Peters95, Dahmen92]。

- 插值分割策略 (Interpolating Subdivision Schemes)

由于在实际应用中，对建立模型的控制能力非常重要，一些改进的近似分割方法强制地对极限网格上的特殊顶点进行插值 [Nasri91, Halstead93]。但这些算法有很多局限性，如插值条件只在极限模型上可以得到满足等。分段线性插值算法则是一种可处理非规则网格的算法，但对一些应用领域来说，其造型结果不够光滑。Dyn 等提出的 Butterfly Scheme [Dyn90] 适用于任意三角网格模型，插值范围仅限于一个很小的邻域，而且算法及数据结构都很简单。该算法生成的极限函数可以达到 C^1 连续，但在网格中的特殊顶点处不能达到这样的光滑程度。Zorin 则对 Butterfly Scheme 加以改进，通过离散傅立叶变换和特征分析在特殊顶点所在的邻域附加了特殊规则，使分割的极限模型可以达到非常光滑的程度 [Zorin96a]。

尽管网格模型的分割算法的数学原理非常复杂，但算法本身很容易实现，并且数据结构简单，执行效率很高。Zorin 的插值策略形式简单、适用于任意拓扑网格模型。本文提出的基于分割插值的多分辨率模型算法即借鉴了 Zorin 的插值方法。

Zorin 在[Zorin96b]中对他的插值规则的数学原理进行了详细阐述。

3.4.2 由分割插值法生成新点

真实场景通常由曲面组成。由于曲面模型的绘制、计算都比较复杂，人们对曲面进行线性逼近，用多边形网格模型来代替包含曲面的真实场景。此外，获得大规模场景，还可通过离散化的采样工具如深度扫描仪，得到场景的网格模型。这里的网格模型都是曲面模型离散化、线性化的结果。因此，在对网格模型进行简化时，如果将新点置于网格模型拟合曲面上，即通过拟合曲面插值来求取新点，能有效地提高简化效果。

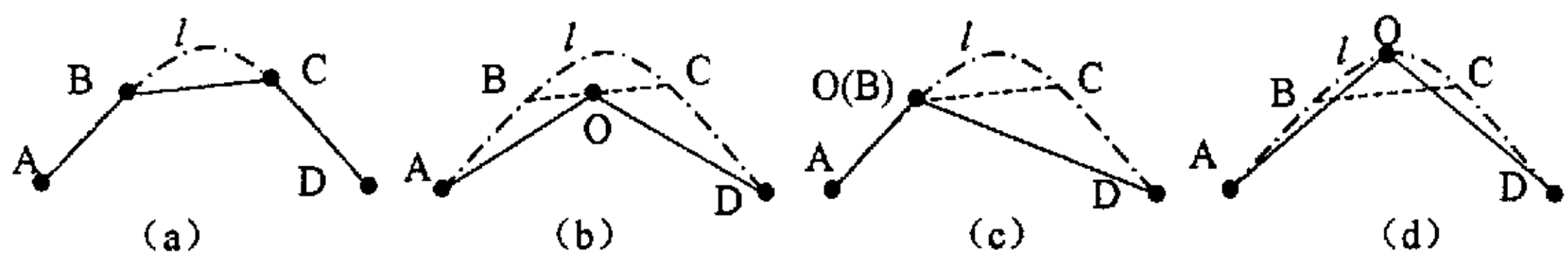


图 3.5 一维情况下边删除操作新点的选取

这里，我们以一维情况为例，介绍采用插值法进行简化的基本思想。如图 3.5 所示，图(a)中点划线为原始曲线 l ，折线 ABCD 为该曲线的线性近似。现对 ABCD 进行简化，删除线段 BC。如图 (b)、(c)、(d) 所示，BC 删除后，新点 O 位置的设定有多种选择。(b) 将 O 置于线段 BC 上一点；(c) 将 O 置于 BC 的端点；(d) 通过插值求取新点。显然，方法 (d) 所得到的简化折线 AOD 是曲线 l 的最好近似。同理，在二维情况，即对网格曲面进行边删除简化时，通过插值求取新点，与简单地利用原始模型顶点或模型边上的点相比，可以更好地得到对模型的近似表示。

求取插值点有多种方法。这里，我们借鉴了 Zorin 的插值方法[Zorin96]。

为了便于讨论我们的插值算法，首先作以下定义：

顶点的价数：顶点的价数为多边形网格中该顶点所邻接的边的数目。

插值模板：对一边 e ，我们取相关区域 $Neighbour(e)$ 内网格顶点的位置矢量的加权平均作为新点的位置矢量。这里，插值模板即为求取插值点时，参与插值的顶点及权值分配的模式。

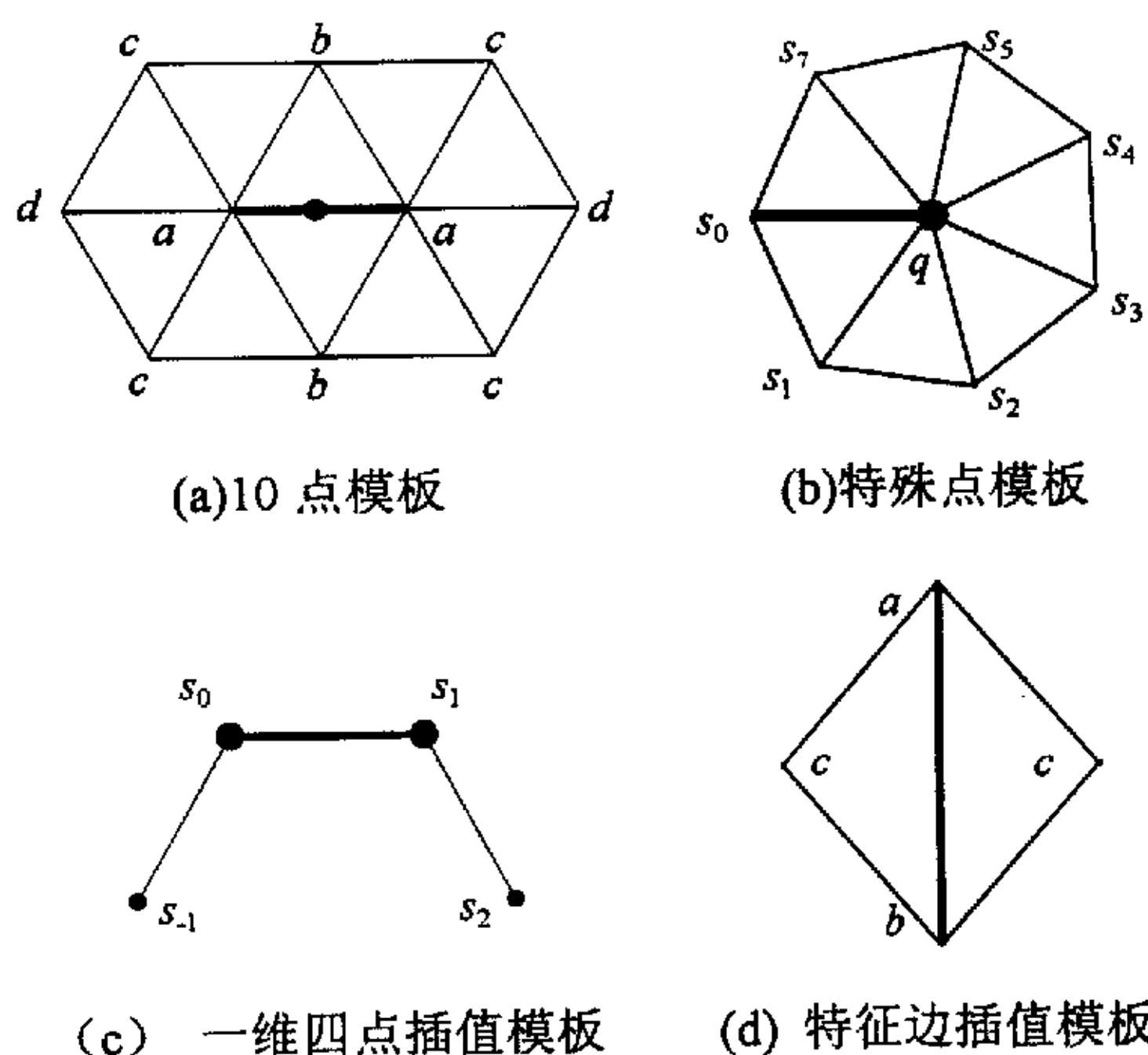


图 3.6 插值模板

如图 3.6 所示为几种插值模板，(a)、(b)、(c)、(d) 中加粗线段表示将删除的边。图中所示 a 、 b 、 c 、 d 、 q 、 $s_i (i = -1, \dots, 7)$ 等符号为计算插值时其所在点的权值。图 3.6 (a) 为两端点的价数均为 6 的边所采用的标准插值模板，即 10 点模板。若边两端点中至少一个价数不为 6，则分别计算两端点的插值点，其模板如图 3.6 (b) 所示（图中由黑点标识的顶点，为当前插值的边端点）。然后对分别得到的两点插值得到新点。图 3.6 (c)、(d) 则分别为边界边、特征边的插值模板。

算法根据边在三角网格模型中不同的拓扑结构及端点重要度，分六种情况求插值点。其中情况 1~3 为两端点重要度均为 0 的边的插值模板；4~6 为两端点重要度均不小于 1 的边的插值模板。

情形 1. 当边两端点价数均为 6 (图 3.6 (a)), 模板中各权值如下定义:

$$a = \frac{1}{2} - w, \quad b = \frac{1}{8} + 2w, \quad c = -\frac{1}{16} - w, \quad d = w$$

这里 w 可取任意小 (我们取 $w=0$)。

情形 2. 当边一端点 6 价, 另一端点 K 价 ($K \neq 6$), 模板定义如图 3.6 (b)。

其中, q 、 s_0 分别为 K ($K \neq 6$) 价端点及 6 价端点的权值。

$K \geq 5$ 时,

$$s_j = \frac{1}{K} \left(\frac{1}{4} + \cos\left(\frac{2\pi j}{K}\right) + \frac{1}{2} \cos\left(\frac{4\pi j}{K}\right) \right) \quad j = -1, \dots, K-1;$$

$K=3$ 时,

$$s_0 = \frac{5}{12}, \quad s_1 = -\frac{1}{12}, \quad s_2 = -\frac{1}{12};$$

$K=4$ 时,

$$s_0 = \frac{3}{8}, \quad s_2 = -\frac{1}{8}, \quad s_1 = 0, \quad s_3 = 0。$$

$$q = 1 - \sum_{i=0, \dots, k-1} s_i。$$

情形 3. 当边的两 endpoint 均为特殊点 (价数 $K \neq 6$), 利用 2 所定义的特殊点模板系数分别计算两端点的模板插值 V_1 、 V_2 , 而新点的位置矢量 $V=V_1+V_2$ 。

情形 4. 当边为模型的边界边时, 通过一维四点插值求点。插值模板如图 3.6(c) 所示:

$$s_{-1} = -\frac{1}{16}, \quad s_0 = \frac{9}{16}, \quad s_1 = \frac{9}{16}, \quad s_2 = -\frac{1}{16}$$

情形 5. 当边两端点重要度均大于 0 时, 插值模板如图 3.6 (d)。其中 a 、 b 、 c 的值根据边两端点的重要度确定 (如表 3.1)。

	V_1 重要度	1	2
V_2 重要度	模板系数		
1		1	3
2		3	2

表 3.1. 边插值模板系数: (1) $a = b = 3/8, c = 1/8$; (2) $a = b = 1/2, c = 0$; (3) $a = 5/8, b = 3/8, c = 0$ (其中, a 为重要度较高的一端点的权值)

情形 6. 当边一 endpoint 重要度为 0, 另一重要度为非 0, 其插值模板如图 3.6 (b)。

其中, $q = 1, s_i = 0 (i = 0, K-1)$ (重要度非零的顶点权因子为 q)。

由以上定义的插值模板可知, 包含 K 个顶点 $V_i (i = 0, \dots, k-1)$, 且各顶点权值为 $s_i (i = 0, \dots, k-1)$ 的模板, 新点 V^* 的位置矢量为

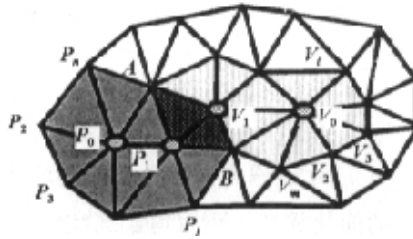
$$V^* = \sum_{i=0, \dots, k-1} s_i V_i \text{。}$$

本章的 3.1~3.4 节, 介绍了本文所提出的基于分割插值的连续多分辨率模型的基本步骤和关键技术。为了进一步提高绘制速度, 我们对算法进行了优化。

3.5 对算法的优化

我们的方法是一种基于边删除的模型简化算法。首先对模型中可删除的边按控制函数值的大小排序, 建立待删除边的队列, 函数值最小的边位于队首。依次取出位于队首的边, 合并边的两 endpoint, 用边相邻区域的光滑插值点作为合并后生成的新点。执行一次边删除操作后, 重新计算其影响区域内的各边的控制函数值, 并调整这些边在队列中的位置。在简化的过程中, 通过改变控制函数中的各分量权值, 使模型在不同的简化阶段, 体现不同的简化原则, 实现理想的简化效果。通过插值求取新点, 有利于维持最小的 Hausdorff 距离, 较好地保持原模型的形状特征。

在算法执行的过程中, 由于每进行一次简化操作都要重新计算影响区域内的各边的控制函数、调整这些边在队列中的位置。显然不同简化操作的影响区域可能存在交集 (如图 3.7), 因而会对交集内的边进行多次计算。如图 3.7 所示, 若对图中边 $(P_0 P_1)$ 执行折叠, 新点为 P^* , 则需要重新计算影响区域内的边 $(P^* P_j) (j=2, 3, \dots, n)$ 的折叠目标点及简化误差, 并把它们冻结, 其中包括边 $(P^* V_1)$ 、 $(P^* A)$ 、 $(P^* B)$ 。假设边 $(V_0 V_1)$ 的折叠操作在该简化操作之后执行, 新点为 V^* 。同样, 算法对该简化操作影响区域内的边 $(V^* V_i) (i=2, 3, \dots, m)$ 的折叠目标点及简化误差进行计算, 并把这些边冻结, 其中边 $(P^* V_1)$ 、 $(P^* A)$ 、 $(P^* B)$ 与前一简化操作所冻结的边相同。由以上分析可以看出, 算法对冻结的边 $(P^* V_1)$ 、 $(P^* A)$ 、 $(P^* B)$ 的折叠目标点及误差进行了多次计算。

图 3.7 边 (P_0P_1) 和 (V_0V_1) 的影响区域

为了消除这种算法上的冗余，我们对算法进行了优化。在 3.3 节我们提到，本文提出的多分辨率模型算法采用了独立集，以避免局部区域的过度简化。可以看出，执行一次简化操作，其影响区域内的边都会被冻结，在被解冻之前不再参加简化，即不会对其它边简化操作的排序产生影响。因此，执行简化操作之后，我们可以不立即处理简化操作影响区域内的边，而直接对队列中下一个保持活跃状态的边进行简化。然后在解冻边的同时，重新计算它们的控制函数值，并进行排序。下面为经优化的算法步骤：

1. 根据模型在各顶点处的曲率，定义各点的重要度。
2. 根据各边两 endpoints 的重要度，判断是否可删除。对可删除的边，计算边删除后生成新点的位置。并求取该边的控制函数值，作为边简化的误差值。
3. 根据各边简化的误差值，对模型中的各边排序，建立待删除边的队列。误差值最小的列于队首。
4. 从待删除队列中取出队首边。将其从网格中删除，以在 (2) 中计算所得的点作为新点，并将该边影响区域内的边冻结。
5. 若队列中所有的边都处于冻结状态，则将队列中的边解冻，重新计算边对应的目标点以及简化误差，并对它们排序、建立队列，误差值最小的列于队首。
6. 当模型简化到一定的程度，交互调整简化控制函数的三个分量的权值，重新计算待简化边队列中各边的折叠误差，执行 (3)。
7. 递归执行 (4)、(5)、(6)，直到简化至用户定义的面片数。

经过优化的算法消除了执行过程中的重复计算，执行效率得到了改善。

3.6 实验结果

该算法已在 Pentium 350 上用 C 语言实现，并成功地应用于不同的模型。利用本文提出的基于分割插值的简化算法，可以得到连续的多分辨率模型。图 3.8、图 3.9 分别列举了算法对球和兔子模型简化到不同程度的效果示例。由实验结果可知，算法可以使模型得到高度的简化（图 3.9）。即使对原始分辨率较低的模型简化到百分之四，仍能较好地保持的原模型的形状及几何特征（图 3.8）。表 3.2、表 3.3 分别列举了图 3.8、图 3.9 多分辨率模型的 Hausdorff 误差。由表可知，该算法成功地减小了简化中的 Hausdorff 误差。

3.7 小结

本章提出了一种基于插值的连续多分辨率模型生成算法。算法通过递归地执行边删除操作，实现对原模型的不简化。边删除的新点由相应模板内的点插值得到，从而有效地控制了 Hausdorff 距离误差。此外，文章提出的可调简化控制函数，使模型以较优的顺序对模型中的各边进行删除简化，使简化网格能更好地保持原模型的形状特征。算法执行的是增量式简化策略，因此可以很方便地以其定义渐进网格模型。我们还可以探索更优化的方法求解插值点。

表 3.2 球的多分辨率模型 Hausdorff 误差 E_{\max} (模型 Hausdorff 误差与模型包围盒对角线的百分比)

绘制模型(简化程度)	面片数	顶点数	E_{\max}
图 3.8 (a)	760	382	0
图 3.8 (b)(50%)	378	191	0.112481
图 3.8 (c)(25%)	188	96	1.163158
图 3.8 (d)(12%)	92	48	2.408543
图 3.8 (e)(6%)	44	24	9.592330
图 3.8 (f)(3%)	20	12	9.592330

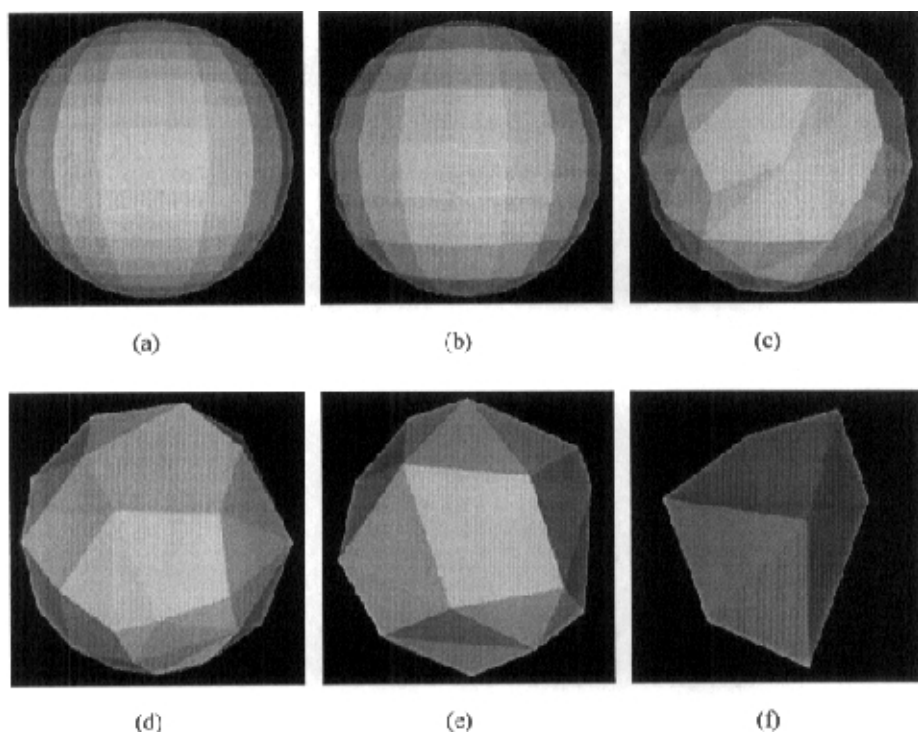


图 3.8 球的多分辨率模型

表 3.3 兔子的多分辨率模型 Hausdorff 误差 E_{\max} (模型 Hausdorff 误差与模型包围盒对角线的百分比)

绘制模型(简化程度)	面片数	顶点数	E_{\max}
图 3.9 (a)	69473	34835	0
图 3.9 (b) (50%)	34716	17417	0.000982
图 3.9 (c) (25%)	17345	8708	0.002703
图 3.9 (d) (10%)	6925	3484	0.007297
图 3.9 (e) (5%)	3464	1746	0.009577
图 3.9 (f) (1%)	687	348	0.021394
图 3.9 (g) (0.5%)	338	173	0.034487
图 3.9 (h) (0.25%)	172	90	0.153557

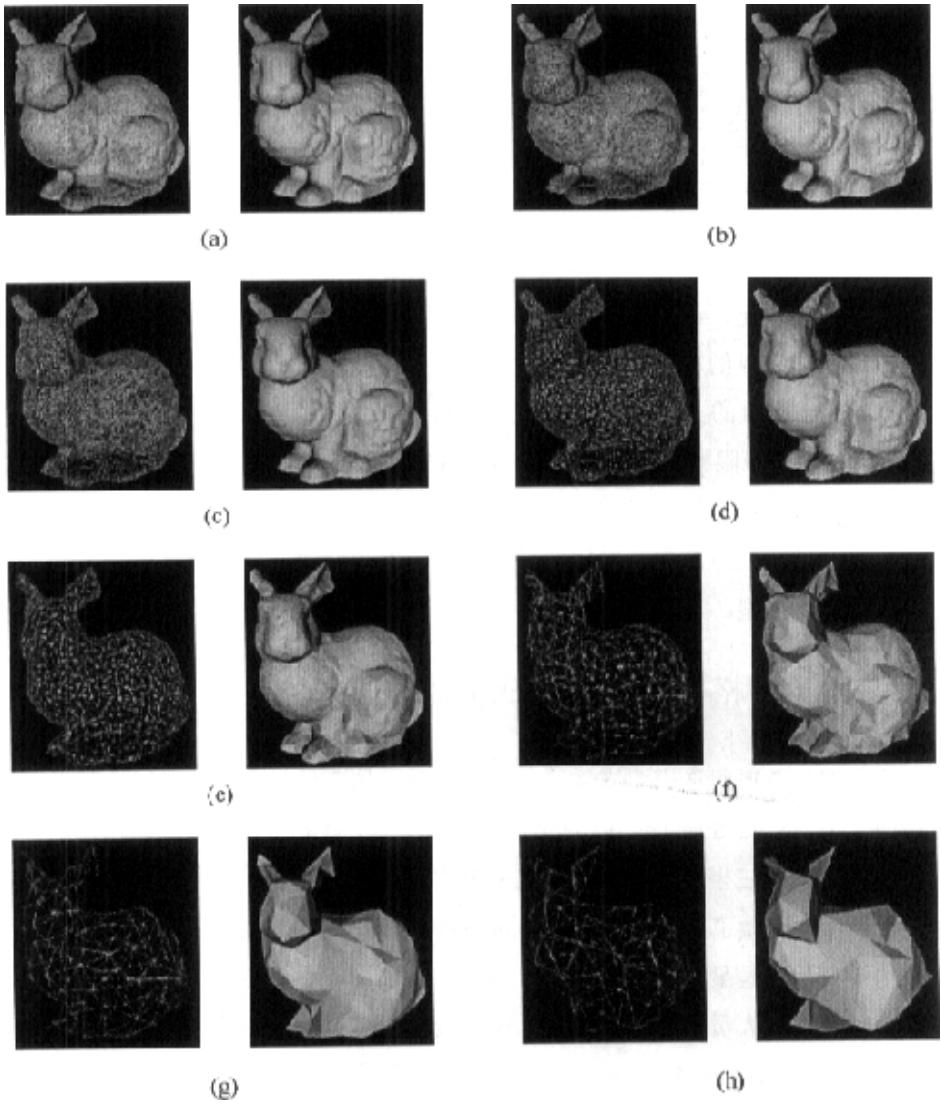


图 3.9 兔子的多分辨率模型

第四章 基于隐函数插值的连续多分辨率模型

众所周知, 场景本身通常是由曲面组成, 而目前应用非常广泛的网格模型则是曲面模型离散化、线性化的结果。本文在第三章讨论的基于分割插值的多分辨率模型, 由相邻点的加权平均位置作为边折叠目标点, 使简化模型更贴近原始模型, 即它们之间的 Hausdorff 距离较小。原始模型插值分割 (Interpolation Subdivision) 的极限模型为原始模型的插值曲面模型, 因此, 实际上, 基于分割插值的算法是通过对原始模型的插值曲面模型的重新采样, 得到简化模型。本章则利用变分插值的方法求得原始模型的插值隐函数曲面, 并通过对该曲面的重新采样得到简化模型。

4.1 三角形网格模型的变分插值曲面

我们把三角形网格模型恢复成曲面, 实际上是希望找到经过给定插值点的代数曲面 (Algebraic Surface) 以便在模型简化过程中计算新点的位置。代数曲面是一种由包含点的三坐标变量的多项式来表达的隐函数曲面。至今为止, 人们已经在代数曲面方面做了很多研究工作[Taubin93]。其中, 如何建立满足给定点集插值约束的代数曲面是最受关注的研究方向。一般来说, 不可能实现对所有已知点的插值, 因此研究人员通常致力于建立误差最小的插值函数。此外, 用很多分离的代数曲面片 (Algebraic Surface Patches) 缝合成曲面模型是这一领域另一研究热点 [Bloomenthal97]。这种方法可以方便地控制对散乱点集的插值, 但它同时存在面片边界的光滑连结难题。

本章采用变分隐函数曲面 [Turk99] 插值网格模型各顶点。

● 变分隐函数曲面

为方便起见, 我们通过介绍变分隐函数曲线在 2D 空间的表达形式——变分隐函数曲线 (图 4.1), 来说明变分隐函数曲面的基本思想。如图 4.1 所示, 曲线由四

个0值点约束（由小空心圆标识）和一个正值点（由加号标识）约束所定义。在0值点，曲线的值为0；正值点，曲线的值为1（可定义为任意正值）。把各约束点的位置及取值作为约束条件，可求得光滑的2D插值函数。所求曲线即为函数值为0的点的轨迹，且曲线内部函数值为正，外部为负。在求取插值函数的过程中，采用变分技术使所生成的函数平均曲率最小。这种方法也被称作薄板插值（Thin-plate Interpolation）。与2D情况类似，我们可以通过定义3D空间点作为0值约束，并在曲面内部的一个或多个点处定义正约束，然后用变分插值技术在3D空间生成标量函数。该函数所有0值点的集合即为变分隐函数曲面。

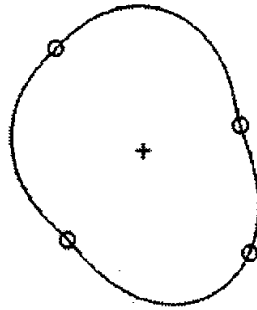


图 4.1 变分隐函数曲线

由以上叙述可知，这种变分插值函数需要通过薄板插值求解，二者有着密切的联系。

● 薄板插值（Thin-Plate Interpolation）

薄板插值因类似于一水平放置的金属薄板被弯曲以接触若干具有一定高度的垂直柱体的顶端的问题而得名。它定义能量 E 来模拟金属板在各个柱体之间光滑过渡的弹力。设在二维情况下，已知 k 个 xy 平面上的约束点 $\{C_1, C_2, \dots, C_k\}$ 以及各点处的高度值 $\{h_1, h_2, \dots, h_k\}$ ，我们可以用薄板插值的方法建立通过各约束点的光滑曲面。实际上，可以把这一曲面看作满足

$$f(C_i) = h_i \quad (1 \leq i \leq k) \quad (4.1)$$

的标量函数 $f(\mathbf{x})$ 。显然，函数 $f(\mathbf{x})$ 有多解。如果我们用能量 E 对“光滑”予以特殊的定义，则可得到唯一解——点的薄板插值。能量函数

$$E = \int_{\Omega} f_{xx}^2(P) + 2f_{xy}^2(P) + f_{yy}^2(P) \quad (4.2)$$

表示定义域 Ω 内曲面 $f(\mathbf{x})$ 的平均曲率。表面上的皱折会导致能量值 E 升高，因此，无高曲率区域的光顺曲面会有较小的能量值。薄板插值就是建立一个满足所有给定约束条件并使 E 最小即曲面平均曲率最小的函数 $f(\mathbf{x})$ 。这里，曲率最小成为插值曲面的附加约束。求解这一约束的方法为变量积分 (Variational Calculus)。因此薄板技术也被称作变分技术 (Variational Techniques)。薄板插值显然可以推广到任意高维的空间，但不同维数空间需要定义不同的能量函数。

● 放射基函数 (Radial Basis Function)

我们可以采用多种数值方法求解上述薄板插值函数。其中，最为常用的有限元 (Finite Element) 以及有限差分 (Finite Differencing Element) 技术把定义域分割成一系列小的单元并在单元内定义局部基函数，函数 $f(\mathbf{x})$ 则被定义为这些基函数的线性组合。有限元和有限差分技术可以用于解决很多领域的问题，并已被广泛应用于高度场插值以及变形模型 [Terzopoulos88, Szeliski90, Celniker91, Welch94]，但其离散性会导致曲面的变形。以约束点为中心的放射基函数 (Radial Basis Function) 是求解薄板插值函数的另一种方法。由于放射基具有关于一点中心对称的性质，已被广泛应用于求解插值函数。显然，我们可以通过适当地选取放射基函数使其能自动求解微分方程并满足各基函数中心点的约束。在二维情况下，求解式 (4.2) 可以采用如下的放射基函数：

$$\phi(x) = |x|^2 \log|x| \quad (4.3)$$

该函数通常作为薄板放射基函数。三维情况下，放射基函数为

$$\phi(x) = |x|^3 \quad (4.4)$$

选用适当的放射基函数 $\phi(x)$ ，我们可以把由 k 个约束 $\{c_1, c_2, \dots, c_k\}$ 定义的插值函数表示为

$$f(x) = \sum_{j=1}^k d_j \phi(x - c_j) + P(x) \quad (4.5)$$

其中 d_j ($j = 1, 2, \dots, k$) 为权值； $P(\mathbf{x})$ 为一次多项式，是函数 $f(\mathbf{x})$ 的线性及常数部分。求解上式中满足约束条件的权值 $\{d_1, d_2, \dots, d_n\}$ 及一次多项式 $P(\mathbf{x})$ 的系数 $\{p_0$

$p_1, p_2, p_3\}$, 可得到经过所有插值点且使能量 E (式 4.2) 最小的函数。显然, 权值的数量决定于约束条件的数量。

求解权值 d_j 需要满足插值约束:

$$h_i = f(c_i), \quad i = 1, 2, \dots, k \quad (4.6)$$

将式 (4.6) 带入式 (4.5) 可得

$$h_i = f(c_i) = \sum_{j=1}^k d_j \phi(c_i - c_j) + P(c_i), \quad i = 1, 2, \dots, k \quad (4.7)$$

上式为未知数权值 d_j ($j = 1, 2, \dots, k$) 以及 $P(x)$ 的四个系数 $\{p_0, p_1, p_2, p_3\}$ 的一次方程组, 可以表达为线性系统 (Linear System)。对于 3D 空间插值, 令 $c_i = (c_i^x, c_i^y, c_i^z)$, $\Phi_{ij} = \Phi(c_i - c_j)$, 则该线性系统为:

$$\begin{bmatrix} \phi_{11} & \phi_{12} & \dots & \phi_{1k} & 1 & V_1^x & V_1^y & V_1^z & d_1 \\ \phi_{21} & \phi_{22} & \dots & \phi_{2k} & 1 & V_2^x & V_2^y & V_2^z & d_2 \\ \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \phi_{k1} & \phi_{k2} & \dots & \phi_{kk} & 1 & V_k^x & V_k^y & V_k^z & d_k \\ 1 & 1 & \dots & 1 & 0 & 0 & 0 & 0 & p_0 \\ V_1^x & V_2^x & \dots & V_k^x & 0 & 0 & 0 & 0 & p_1 \\ V_1^y & V_2^y & \dots & V_k^y & 0 & 0 & 0 & 0 & p_2 \\ V_1^z & V_2^z & \dots & V_k^z & 0 & 0 & 0 & 0 & p_3 \end{bmatrix} = \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_k \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (4.8)$$

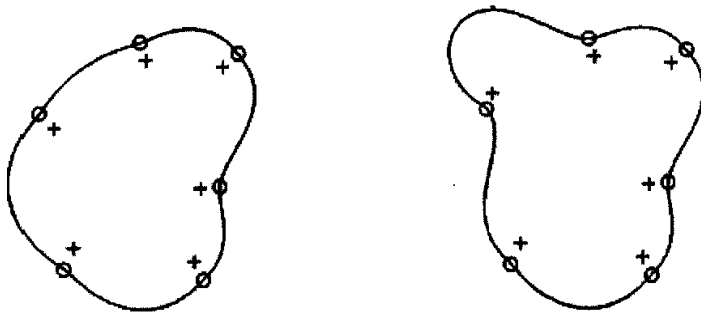
可以证明, 式 (4.8) 中矩阵为对称半正定, 因此, 其中的 d_j ($j = 1, 2, \dots, k$) 和 p_i ($i = 0, 1, 2, 3$) 存在唯一解 [Golub96], 并可用数值方法求解。

● 生成变分隐函数曲面:

生成变分隐函数曲面, 除了定义 0 值约束 (边界约束——Normal Constraints) 之外, 还需要定义正值约束。本文生成变分隐函数曲面的目的在于插值一个给定的多边形网格模型 S 。而网格模型本身包含丰富的几何信息, 如各顶点位置矢量以及各顶点的法向等。设网格模型的顶点 P 处函数值为 0, 则定义了所求变分隐函数曲面的边界约束 (Boundary Constraints)。此外, 网格模型在顶点 P 处的法向 N_i 可以近似表示对应隐函数曲面在同一位置的法向, 我们可以利用该法向信息在接近边界约束的位置生成约束, 如定义点 $P_i + hN_i$ 处 (h 为一小负值如 -0.01) 的函数值为正值, 作为法向约束 (Normal Constraints)。显然法向约束反映了正值点的信息。如图 4.2 所示为二维空间中, 由边界约束及法向约束定义的变分隐函数曲线。

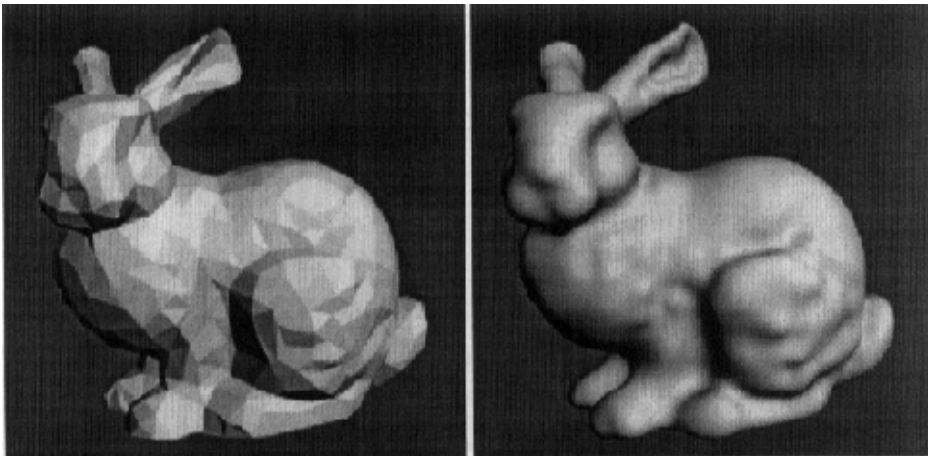
$$f(C_i) = h_i = \begin{cases} 0 & \text{当 } C_i = P_i \\ h & \text{当 } C_i = P_i + hN_i \end{cases} \quad i = 1, 2, \dots, m \quad (4.9)$$

由上述约束条件，可以通过求解式4.8得到满足约束条件的3D空间标量函数 $f(P) (P \in R^3)$ ——变分隐函数曲面。



(a) 由边界约束及法向约束定义的曲线 (b) 修改左上角顶点法向约束后的曲线

图 4.2 两个由几乎相同的边界及法向约束定义的曲线



(a) 多边形网格模型

(b) 变分隐函数曲面模型

图 4.3 兔子多边形网格的变分插值隐函数曲面

由于采用了使曲面平均曲率最小的放射基函数, 变分隐函数曲面可以严格地插值给定的散乱点集, 且表面较为光滑, 适用于描述任意拓扑曲面。变分隐函数曲面的一个主要优点在于, 它不仅可以满足曲面上点的位置约束, 还能满足曲面的法向约束。图 4.3 (b) 所示为由图 4.3 (a) 所示多边形网格模型使用上述方法所得到的变分隐函数曲面。

4.2 基于变分隐函数插值的连续多分辨率模型

本章所讨论的基于变分隐函数插值的连续多分辨率模型是一种基于边删除操作的模型简化算法。算法以原始网格模型为初始模型, 通过递归地执行边删除操作 (本文 3.1 节), 得到原始模型的简化模型。

算法在执行简化操作之前, 首先用变分插值技术生成原始网格模型的变分隐函数曲面。在执行简化的过程中, 通过对曲面的采样求得边折叠的目标点。算法的基本步骤如下:

1. 生成网格模型的变分插值曲面。
2. 建立待删除边集合。
3. 通过采样变分插值曲面计算待删除边集合内各边对应新点的位置, 并求取该边的简化误差。
4. 根据简化误差对待删除边集合内的边排序, 建立队列。误差值最小的边列于队首。
5. 从待删除边队列中取出位于队首的边。将其从网格中删除, 用在 (3) 中计算所得的新点代替该边。
6. 重新计算相关边对应的新点以及简化误差, 调整其在队列中的位置
7. 递归执行 (5)、(6), 直到简化至用户定义的面片数, 或达到给定的简化误差阈值。

其中, 生成网格模型的变分隐函数插值曲面, 是通过数值方法求解式 (4.8) 得到的, 矩阵的维数只与插值顶点的数目 m 有关, 为 $2 \times m + 4$ 。显然, 求解这样的矩阵系统, 需要考虑计算时间和病态方程 (Ill-conditioned Systems)。如果我们对整个网格模型的变分隐函数插值生成曲面模型, 对于包含几千个点的较为简单的模

型，可以采用 LU 分解等方法直接求解方程组。但由于求解方程所需要的计算量为 $O(m^3)$ ，当模型复杂度提高时，计算量增大的很快。当然可以采用计算效率更高的方法求解方程[Suter94, Beatson97]来提高计算速度。但是，随着约束的增多，会使矩阵的条件数（Condition Number）增大，从而导致求解过程的不稳定。因此，我们需要首先分割原始模型，并在各个独立的局部区域内求解网格模型的变分隐函数插值曲面（4.2.1）。

此外，在简化过程中，算法沿用第三章所述的可调加权控制函数（本文 3.3）估计简化误差，在此不作赘述。

4.2.1 隐函数插值的局部网格模型

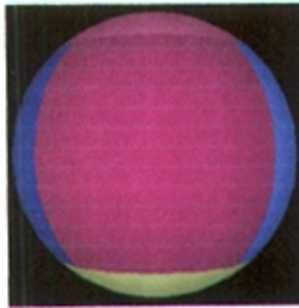
一种比较方便的局部区域变分插值方法，就是在网格模型上各条边的影响区域内（图 3.1(a)）进行插值，并通过对该插值曲面的采样确定这条边的折叠目标点。这种方法简单，容易实现。但是，大量的小曲面片的拼合显然会损害变分隐函数曲面的光顺等良好的性质。同时，曲面的生成依赖于模型上的顶点的位置矢量以及法向，由于执行边删除简化操作后需重新计算相关边所对应的插值曲面，会导致插值曲面片与原始模型所对应曲面模型的相关性大大降低，进而丧失对边折叠目标点计算的可参考性。

因此，我们选择基于高斯球面网格对网格模型进行分割（本文第二章）生成局部插值区域的策略，在减小算法计算复杂度的同时在一定程度上保持了网格模型所对应曲面模型的完整性。并且，区域插值曲面在模型简化过程中保持不变，不会因简化而产生累积误差，使其上的采样点对边折叠目标点计算的可参考性恒定在最高值。注意到，位于区域边界的边同时对应于两个以上的区域插值曲面。这些曲面之间不存在连续性约束，区域插值曲面的简单拼合会导致折痕出现。因此，我们对位于区域边界的边单独建立边局部区域的插值曲面。

这样，基于区域隐函数插值的多分辨率模型简化算法在执行简化操作之前，首先基于高斯球面网格分割网格模型成插值区域，并用变分插值的方法对区域内的模型顶点插值生成分区域的变分插值曲面。对位于区域边界的边，建立边局部区域的插值曲面。在算法执行简化的过程中，通过对这些插值曲面的采样求得边

折叠的目标点。即：

1. 分割网格模型，建立网格模型分片插值区域（图 4.4）。
2. 生成各插值区域的变分插值曲面，把所得的区域插值函数作为区域内边对应的函数。对于区域边界边，单独计算边相关区域（图 3.1）的插值曲面，并以此作为该边的对应函数。
3. 建立待删除边集合。
4. 通过采样相应的变分插值曲面（区域内边所对应插值曲面为区域插值函数；区域边界边所对应插值函数为边局部区域插值函数。）计算待删除边集合内各边对应新点的位置，并求取该边的简化误差。
5. 根据简化误差对待删除边集合内的边排序，建立队列。误差值最小的边列于队首。
6. 从待删除边队列中取出位于队首的边。将其从网格中删除，用在（4）中计算所得的新点代替该边。
7. 若相关边为区域边界边，重新计算边对应的插值函数；否则，转（8）；
8. 重新计算相关边的对应的新点简化误差，调整其在队列中的位置
9. 递归执行（6）、（7）、（8），直到简化至用户定义的面片数，或达到给定的简化误差阈值。



(a)分区域的球面模型



(b)分区域的牛模型

图 4.4 建立模型的分片插值区域

4.2.2 新点计算

基于隐函数插值的多分辨率模型算法，通过对边所对应的插值函数的采样求取该边删除操作的折叠目标点。我们可以通过一些复杂的优化方法求得插值曲面在局部区域内的最优采样点，但优化算法需要较多的计算时间。另一方面，由于变分隐函数曲面具有光顺的性质，我们可以利用一些简单的策略对曲面采样取得近似优化的结果。

设待简化边为 V_1V_2 ， $M = (V_1+V_2) / 2$ 为其中点；边相邻区域内（图 3.1(a)）的点 $\{V_1, V_2, V_3, V_4, \dots, V_k\}$ 的平均位置点 $C = \frac{1}{k} \sum_{i=1}^k V_i$ 。把点 M 或 C 沿法矢量 N 向相关插值曲面上投影的方法进行采样，并作为边的折叠目标点。其中，点 M 的投影法向取为边 V_1V_2 相邻面片的平均法向；点 C 的投影方向取为边相关区域内所有面片的法向的平均值。

我们采用了二分法计算投影点的坐标。

- 计算上述投影点的坐标，即为求经过给定点 V ，沿给定方向 N 的直线与隐函数曲面 $f(x)$ 的交点。采用二分法可以很方便地求解这一问题：

1. 令 $M_0 = M$;
2. 以 M_0 为原点，沿 N 移动步长 l ，得 M_1 ，其中 $N = \begin{cases} N & f(M) > 0 \\ -N & f(M) < 0 \end{cases}$;
3. 若 $f(M_1) = 0$ ，则 M_1 为所求交点，结束计算；若 $f(M_1)$ 的符号与 $f(M_0)$ 相同，则令 $M_0 = M_1$ ，执行 (2)；若 $f(M_1)$ 的符号与 $f(M_0)$ 相反，则表明交点在点 M_1 及 M_0 之间，执行 (4)；
4. 令 $P_0 = M_0$ ， $P_1 = M_1$ ；
5. 若 $|P_0 - P_1| \leq \varepsilon$ (ε 为所定义阈值)，则取 $(P_0 + P_1) / 2$ 为交点，结束计算；否则，执行 (6)；
6. 令 $P = (P_0 + P_1) / 2$ 。若 $f(P) = 0$ ，则点 P 为所求交点，结束计算；若 $f(P)$ 与 $f(P_0)$ 符号相同，则令 $P_0 = P$ ；若与 $f(P_1)$ 符号相同，则令 $P_1 = P$ ；执行 (5)；

4.3 实验结果

我们已在 Pentium350 上用 C++语言实现了上述算法, 并对牛、河马、球面等多组网格数据进行了检测。实验结果表明, 用该算法所生成的简化模型可以很好地保持原始模型的几何特征, 并且维持较小的 Hausdorff 误差。

图 4.5、4.6、4.7 为本算法的实验结果。图 4.5 中, (a) 为河马的原始网格模型, 有 5094 个面片、2555 个顶点; (b) 为原始模型的顶点数简化至 25%后的模型, 由 1262 个面片、639 个顶点组成。(c)、(d) 分别为 (a)、(b) 所示模型的面表示。图 4.6 为牛的多分辨率模型。其中 (a) 为牛的原始模型, 包含 3107 个顶点、6208 个面片。(b)、(c)、(d)、(e)、(f) 均为 (a) 的简化模型, 其细节层次依次降低, 分别简化了原始模型的 50%、75%、88%、94%、98.5%。图 4.7 为算法对球模型的简化结果。表 4.1、4.2 分别为对应于图 4.6 和图 4.7 中牛和球模型的不同分辨率模型的 Hausdorff 误差。与第三章的实验结果(图 3.8、表 3.2)比较可知, 由于采用了对原始模型的插值隐函数曲面重新采样的方法进行模型简化生成多分辨率模型, 获得了较好的简化效果。由图可见, 即使高度简化的模型也能较好地保持的原模型的形状及几何特征。

4.4 小结

本章讨论了一种基于隐函数插值的分辨率模型生成算法。考虑到目前应用十分普遍的网格模型实际上是曲面模型离散化、线性化的结果, 我们在对网格模型进行简化之前, 首先生成网格模型的插值隐函数曲面。与第三章所提出的基于分割插值的连续多分辨率模型算法类似, 新方法通过递归地执行边删除操作, 实现对原模型的增量式简化, 生成连续多分辨率模型。在简化的过程中, 以该表面上的采样点作为边删除后的新点, 从而有效地减小了简化所带来的 Hausdorff 误差。该算法比第三章算法优越之处在于, 它对原始网格模型所生成的隐函数插值曲面始终作为简化过程中计算新点位置的依据, 从而避免了累积误差, 使简化网格能更好地保持原模型的形状特征。

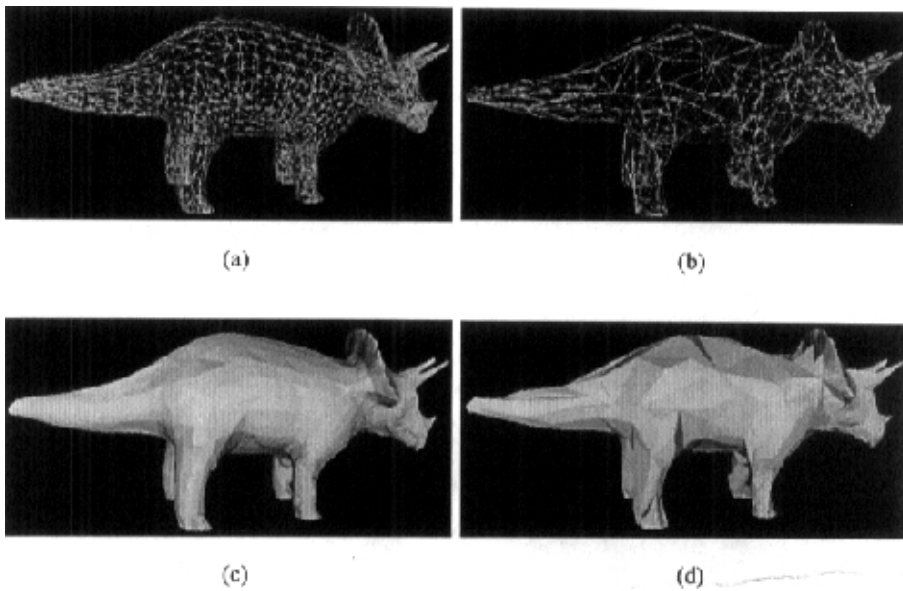


图 4.5 河马模型的简化

表 4.1 牛的多分辨率模型 Hausdorff 误差 E_{max} (模型 Hausdorff 误差与模型包围盒对角线的百分比)

绘制模型(简化程度)	面片数	顶点数	E_{max}
图 4.6 (a)	6208	3107	0
图 4.6 (b) (50%)	3102	1554	0.630411
图 4.6 (c) (25%)	1548	777	1.763023
图 4.6 (d) (12%)	774	390	2.735106
图 4.6 (e) (6%)	390	198	3.264701
图 4.6 (f) (1.5%)	88	46	10.36134

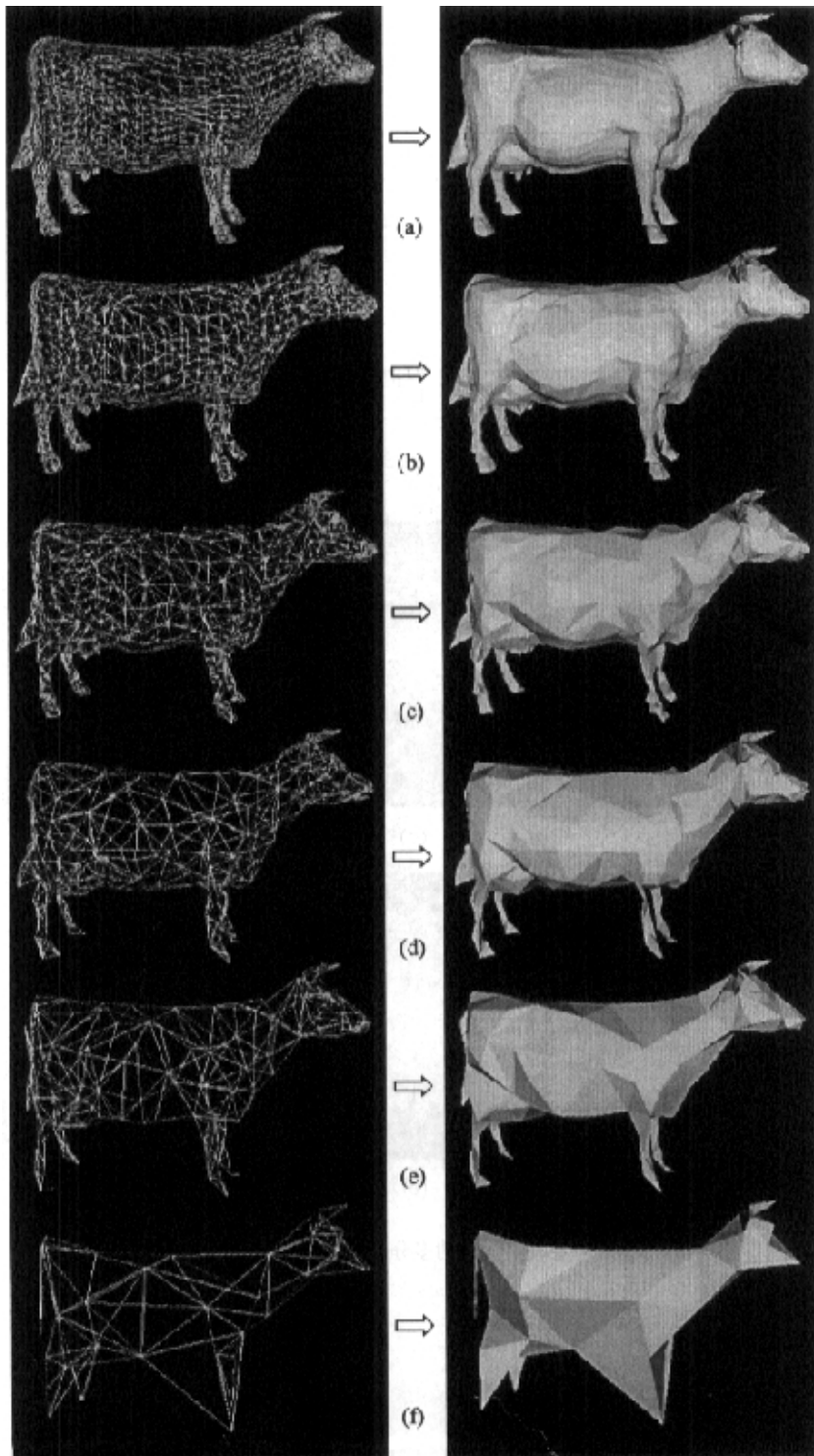


图 4.6 牛的多分辨率模型

表 4.2 球的多分辨率模型 Hausdorff 误差 E_{\max} (模型 Hausdorff 误差与模型包围盒对角线的百分比)

绘制模型(简化程度)	面片数	顶点数	E_{\max}
图 4.7 (a)	760	382	0
图 4.7 (b)(50%)	378	191	0.067503
图 4.7 (c)(25%)	188	96	0.122463
图 4.7 (d)(12%)	92	48	0.795842
图 4.7 (e)(6%)	44	24	2.259140
图 4.7 (f)(3%)	20	12	9.592330

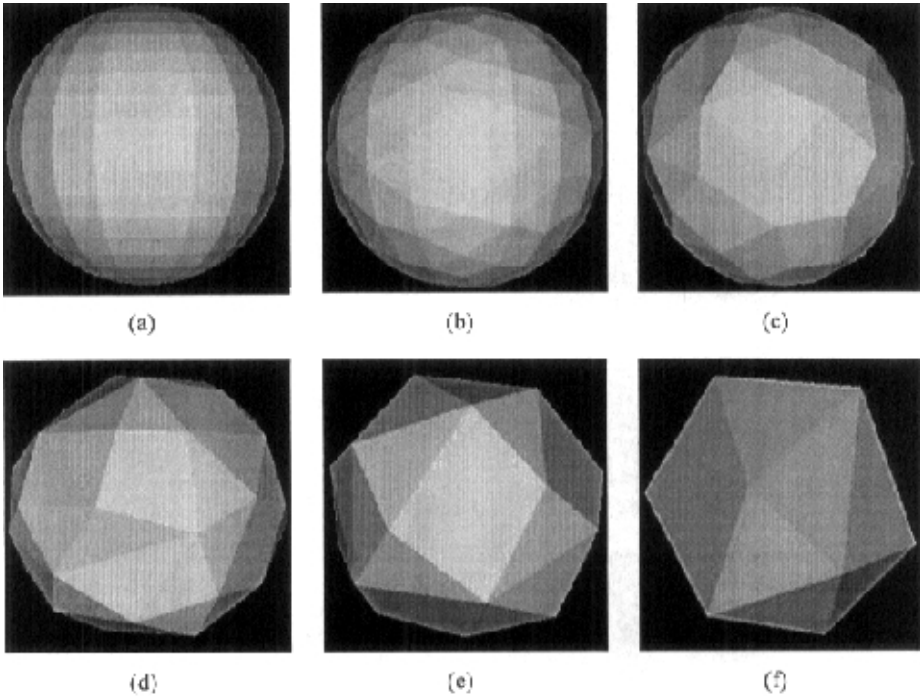


图 4.7 球的多分辨率模型

第五章 总结

本文主要对保持几何特征的复杂景物多分辨率模型技术进行了研究和讨论。不仅系统地讨论了现有的多分辨率模型技术，而且提出了基于高斯球的近似共面合并多分辨率模型、基于插值分割的多分辨率模型以及基于隐函数插值的多分辨率模型等一系列新的多分辨率模型。本章将总结本论文的工作，并提出对今后工作的展望。

5.1 论文工作的总结

多分辨率模型，是一种对场景对象的多层次表示方法。它通过建立原始场景模型的多个近似简化模型，以不同程度的细节来表示原始模型。随着 VR 技术的发展，实时绘制成为人们越来越关心的问题。但由于计算机硬件水平与场景模型的无限复杂性之间的矛盾关系，需要采用一些加速手段来解决。应用多分辨率模型技术，系统在实时绘制过程中，对离视点较远的视觉重要度较低的场景部分选用简化的模型，从而减少系统需绘制的面片数目，提高绘制速度，达到实时显示的目的。

到目前为止，人们在多分辨率模型技术方面已经做了很多有意义的研究工作，提出了一些有效的算法。本文在总结前人工作的基础上，在多分辨率模型技术方面进行了有益的探索，主要解决了以下问题：

- 提出了基于高斯球面层次分割的近似共面合并多分辨率模型。基于近似共面合并的模型简化算法在保持模型几何特征方面具有其它算法无法比拟的优越性。但由于算法复杂，不易实现，人们在这方面的研究较少。本文提出的算法通过对高斯球面的层次分割，实现对模型的简化以及多分辨率模型的生成。算法简单，易于实现，计算速度很快，并且可以有效地保持模型的几何特征。

- 提出了基于分割插值的连续多分辨率模型。该算法是通过递归地执行边删除操作来实现对模型的简化的。考虑到我们所处理的许多网格模型是曲面模型离散化的结果，本文基于分割插值的原理，试图通过相邻顶点的插值使边折叠目标点更靠近网格模型所对应的曲面，从而获得较小的模型近似误差。此外，我们还采用了可调权值的控制函数估计边删除简化操作的误差，对模型的简化顺序进行控制，获得较为优化的简化结果。
- 提出了基于隐函数插值的连续多分辨率模型。与基于分割插值的连续多分辨率模型相同，这一算法也是基于边删除来进行模型简化。所不同的是，算法采用变分隐函数插值的方法对模型分区域插值，生成原始模型的区域插值隐函数曲面。通过对隐函数曲面的采样，求得边折叠的目标点。由于区域插值隐函数曲面在模型简化过程中保持不变，可以避免简化过程所带来的累积误差。

5.2 今后的工作

应该说，我们所做的工作离理想目标还有很大差距，仍有很多问题有待解决：

- 基于高斯球的近似共面合并多分辨率模型算法，可以采用更为鲁棒的三角化算法提高算法的适用范围。
- 本文所提出的基于插值的连续多分辨率模型取对应曲面模型采样点作为简化模型顶点以及采用可调权值控制函数对模型简化的对象进行控制，为模型简化算法提供了新的思路。实际上，我们还可以把这一算法推广到基于点对折叠的简化模型，实现对模型拓扑结构的简化修改。此外，还可以探索更优化的插值方法和控制函数，以得到更高质量的简化模型。
- 我们还可以通过把纹理、颜色、材料等标量性质引入算法中，实现视觉等价的模型简化。
- 对于深度层次差别较大的场景模型，仅采用模型简化的方法还不能实现在保持视觉等价的前提下最大程度地减小系统的绘制量。当景物距离视点较远以致对视觉的影响甚小或仅在屏幕上投影很少的象素，采用最简化的模型仍觉冗余。在这种情况下可利用基于图象的绘制方法，实现高

效绘制。即我们有必要结合基于图象的表示技术把本文提出的基于模型简化的快速绘制方法发展成为基于图形、几何的混合快速绘制技术，提高对大规模复杂场景的绘制速度。

参考文献

- [Abi-Ezzi91] S.S. Abi-Ezzi and L.A. Shirman. Tessellation of curved surfaces under highly varying transformations. Proceedings of Eurographics'91, pp. 385~397, 1991.
- [Abi-Ezzi94] S.S. Abi-Ezzi and S. Subramaniam. Fast dynamic tessellation of trimmed nurbs surfaces. Computer Graphics Forum, 13(3):107~126, 1994. Proceeding of Eurographics'94.
- [Adelson91] Adelson, E. H. and Bergen J. R.. The plenoptic function and the elements of early vision. In M. Landy and J. Movshon, editors, *Computational Models of visual Processing*, chapter 1. MIT Press, Cambridge, MA, 1991.
- [Agarwal94] Pankaj K. Agarwal and Subhash Suri. Surface approximation and geometric partitions. In Proc. 5th ACM-SIAM Sympos. Discrete Algorithms, pp. 24~33, 1994.
- [Airey90] J. Airey, J. Rohlf, and F. Brooks. Towards image realism with interactive update rates in complex virtual building environments. In Symposium on Interactive 3D Graphics, pages 41~50, 1990.
- [Algorri96] María-Elena Algorri and Francis Schmitt. Mesh simplification. Computer Graphics Forum, 15(3), August 1996. Proc. *Eurographics '96*.
- [Ball88] A. A. Ball, D. J. T. Storry, Conditions for tangent plane continuity over recursively generated B-spline surfaces, ACM Transactions on Graphics 7, 2 (1988), 83~102
- [Beatson97] R. K. Beatson and W. A. Light. Fast evaluation of radial basis functions: Methods for two-dimensional polyharmonic splines. *IMA Journal of Numerical Analysis*, 17:343~372, 1997.
- [Blinn76]. Blinn, J. F. and Newell, M. E., Texture and reflection in computer generated images. CACM, October 1976, 19(10), pp. 542~547.

- [Bloomenthal97] Jules Bloomenthal, editor. *Introduction to Implicit Surfaces*. Morgan Kaufmann Publishers, Inc., San Francisco, California, 1997.
- [Bryson94a] Steve Bryson, Developing Advanced Virtual Reality Applications, Siggraph'94 Course #2, 1994
- [Bryson94b] Steve Bryson, Run-time Architectures and Time Management for Unsteady Interactive Visualization Environments, Siggraph'94 Course #2, 1994
- [Cao99a] 曹卫群 鲍虎军 彭群生, 基于高斯球的近似共面合并多层次细节模型, 软件学报 (已录用)
- [Cao99b] 曹卫群 刘新国 鲍虎军 彭群生, 基于分割插值的连续多分辨率模型, 软件学报 (已投出)
- [Catmull74] Catmull, E., A Subdivision Algorithm for Computer Display of Curved Surfaces. PhD thesis. University of Utah, 1974..
- [Catmull78] E. Catmull and J. Clark. Recursively Generated B-Spline Surfaces On Arbitrary Topological Meshes. *Computer Aided Design*, 1978, 10(6): 350~355.
- [Celniker91] George Celniker and Dave Gossard. Deformable curve and surface finite-elements for free-form shape design. *Computer Graphics (SIGGRAPH'91)*, 25(4):257~266, July 1991.
- [Certain96] Andrew Certain, Jovan Popović, Tony DeRose, Tom Duchamp, David Salesin, and Werner Stuetzle. Interactive multiresolution surface viewing. In *SIGGRAPH'96 Conference Proceedings*, pp. 91~98, August 1996.
- [Chaikin74] G. Chaikin. An algorithm for high speed curve generation. *Computer Graphics and Image Processing*, 1974(3): 346~349.
- [Chen93] Chen, S. E. and Williams L. View interpolation for image synthesis. In *Computer Graphics (SIGGRAPH'93)*, pp. 279~288, 1993.
- [Cheng98] 成迟意 潘志庚 石教英, 递进网格的一种快速生成算法, 中国图象图形学报, 3(11):946~951, 1998.
- [Chew93] L. Paul Chew. Guaranteed-quality mesh generation for curved surfaces. In *Proc. Ninth Annual Symposium on Computational*

- Geometry, pages 274~280. ACM, 1993.
- [Ciampalini97] A. Ciampalini, P. Cignoni, C. Montani, and R. Scopigno. Multiresolution decimation based on global error. *The Visual Computer*, 13(5):228~246, 1997.
- [Cignoni98] P. Cignoni, C. Montani, C. Rocchini, and R. Scopigno. A general method for preserving attribute values on simplified meshes. In *IEEE Visualization'98 Conference Proceedings*, pp.59~66,518, Oct 1998.
- [Clark76] Clark James H. Hierarchical Geometric Models for Visible Surface Algorithm. *Communications of the ACM*, 1976, 19(10): 547~554.
- [Cohen97] Jonathan Cohen, Dinesh Manocha, and Marc Olano. Simplifying polygonal models using successive mappings. In *Proceedings IEEE Visualization'97*, pp.395~402, October 1997.
- [Cohen98] Jonathan Cohen, Marc Olano, and Dinesh Manocha. Appearance-preserving simplification. In *Proceedings SIGGRAPH'98*, pp.115~122, 1998.
- [Coorg96] S. Coorg and S. Teller. Temporally coherent conservative visibility. In *Proc. of 12th ACM Symposium on Computational Geometry*, 1996.
- [Cormen90] Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest. *Introduction to Algorithms*. MIT Press, Cambridge, MA, 1990.
- [Cruz_Neira98a] Carolina Cruz-Neira, Overview of Virtual Reality, *Siggraph'98 Course #14*, 1998
- [Cruz_Neira98b] Carolina Cruz-Neira , Making Virtual Reality Useful, *Siggraph'98 Course #14*, 1998
- [Dahmen92] W. Dahmen, C. A. Micchelli, H. P. Seidel, Blossoming Begets B-Splines Based Built Better by B-Patches. *Mathematics of Computation* 59, 199(July 1992), 97~115
- [Dam94] Andries van Dam, VR as a Forcing Function: Software Implications of a New Paradigm, *Siggraph'94 Course #2*,1994
- [Debevec96] Debevec, P. E. , Taylor C. J. and Malik J. Modeling and rendering architecture from photographs : A hybrid geometry-

- and image-based approach. In *Computer Graphics (SIGGRAPH'96)*, pp. 11~20, August 1996.
- [Delingette94] Hervé Delingette. Simplex meshes: a general representation for 3D shape reconstruction. Technical report, INRIA, Sophia Antipolis, France, Mar. 1994.No. 2214
- [Delingette92] Hervé Delingette, Martial Hebert, and Katsushi Ikeuchi. Shape representation and image segmentation using deformable surfaces. *Image and Vision Computing*, 10(3):132~144, Apr. 1992.
- [Doo78] D. Doo and M. A. Sabin. Behaviour Of Recursive Subdivision Surfaces Near Extraordinary Points. *Computer Aided Design*, 1978,10(6):356~360.
- [Duchon77] J. Duchon. Spline minimizing rotation-invariant semi-norms in sobolev spaces. In W. Schempp and K. Zeller, editors, *Constructive Theory of Functions on Several Variables, Lecture Notes in Mathematics 571*, Berlin, 1977. Springer-Verlag.
- [Dyn90] N. Dyn, D Levin, J. A. Gregory, A Butterfly Subdivision Scheme for Surface Interpolation with Tension Control, *ACM Transactions on Graphics* 9, 2(April 1990), 160~169
- [Eck95] Matthias Eck, Tony DeRose, Tom Duchamp, Hugues Hoppe, Michael Lounsbery, and Werner Stuetzle. Multiresolution analysis of arbitrary meshes. In *SIGGRAPH '95 Proc.*, pages 173~182. ACM, August 1995.
- [Eric96] Eric J. Stollnitz, Tony D. DeRose, and David H. Salesin. Wavelets for Computer Graphics: Theory and Applications. Morgan Kaufmann, San Francisco, CA, 1996.
- [Faugeras84] Olivier Faugeras, Martial Hebert, P. Mussi, and Jean-Daniel Boissonnat. Polyhedral approximation of 3-D objects without holes. *Computer Vision, Graphics, and Image Processing*, 25:169~183, 1984.
- [Floriani98] Leila De Floriani, Paola Magillo, and Enrico Puppo. Efficient implementation of multi-triangulations. In *IEEE Visualization '98 Conference Proceedings*, pp.43~50,517, Oct 1998.
- [Fuch94] Fuchs, H. , bishop G , Arthur K., McMillan L. , Bajcsy R. , Lee S. , Farid H. , and Kanade T. Virtual space teleconferencing using a sea of cameras. In *First International Symposium on*

- Medical Robotics and Computer Assisted Surgery*, Pittsburgh, PA, September 1994.
- [Garland95] Michael Garland and Paul S. Heckbert. Fast polygonal approximation of terrains and height fields. Technical Report CMU-CS-95-181, Comp. Sci. Dept., Carnegie Mellon University, September 1995.
- [Garland97] Michael Garland and Paul S. Heckbert. Surface simplification using quadric error metrics. In *SIGGRAPH'97 Proc.*, pp. 209~216, August 1997.
- [Garland98] Michael Garland and Paul S. Heckbert. Simplifying surfaces with color and texture using quadric error metrics. In *IEEE Visualization'98 Conference Proceedings*, pp.263~269,542, October 1998.
- [Ghee96] Steve Ghee, Programming Virtual Worlds, SIGGRAPH '96 Course #14,1996
- [Gieng97] Tran S. Gieng, Bernd Hamann, Kenneth I. Joy, Gregory L. Schlussmann, and Isaac J. Trotts. Smooth Hierarchical Surface Triangulations. *Proceedings of IEEE Visualization'97* pp.379~386, 1997
- [Gieng98] Tran S. Gieng, Bernd Hamann, Kenneth I. Joy, Gregory L. Schussman, and Isaac J. Trotts. Constructing hierarchies for triangle meshes. *IEEE Trans. on Visualization and Computer Graphics*, 4(2):145~161, April~June 1998.
- [Girosi93] Federico Girosi, Michael Jones, and Tomaso Poggio. Priors, stabilizers and basis functions: from regularization to radial, tensor and additive splines. Technical report, MIT Artificial Intelligence Laboratory, June 1993. A.I. Memo No. 1430.
- [Golub96] Gene H. Golub and Charles F. van Loan. *Matrix Computations*. John Hopkins University Press, 1996.
- [Gortler96] Gortler, S. J. , Grzeszczuk R. , Szekiski R. and Cohen M. F. The Lumigraph. In *Computer Graphics (SIGGRAPH' 96)*, pp. 43~54, August 1996.
- [Gourdon95] Alexis Gourdon. Simplification of irregular surface meshes in 3D medical images. In *Computer Vision, Virtual Reality, and Robotics in Medicine (CVRMed '95)*, pp.413~419, April 1995.

- [Greene93] N. Greene, M. Kass, and G. Miller. Hierarchical z-buffer visibility. In Proc. of ACM Siggraph, pp 231~238, 1993.
- [Guéziec95] André Guéziec. Surface simplification with variable tolerance. In Second Annual Intl. Symp. on Medical Robotics and Computer Assisted Surgery (MRCAS '95), pp. 132~139, November 1995.
- [Guéziec96] André Guéziec. Surface simplification inside a tolerance volume. Technical report, Yorktown Heights, NY 10598, March 1996. IBM Research Report RC 20440,
- [DeHaemer91] Michael J. DeHaemer and Zyda, M. J.. Simplification of objects rendered by polygonal approximations. Computers & Graphics, 1991, 15(2), 175~184.
- [Gillilan98] Richard Gillilan, Scientific Applications of Virtual Reality, Siggraph'98 Course #14, 1998
- [Halstead93] M. Halstead, M. Kass, T. Derosé, Efficient, Fair Interpolation using Catmull-clark surfaces, Computer Graphics Proceedings, Annual Conference Series, 35~44, 1993
- [Hamann94] Bernd Hamann, A Data Reduction Scheme for Triangulated Surfaces. Computer Aided Geometric Design. 11(2):197~214, 1994
- [Heckbert94] Paul S. Heckbert and Michael Garland. Multiresolution modeling for fast rendering. In Proc. Graphics Interface '94, pp.43~50, Banff, Canada, May 1994.
- [Hinker93] Hinker, P. and Hansen, C., Geometric optimization. In IEEE Visualization '93 Proc., October, 1993, pp.189~195.
- [Holloway93] Richard Holloway, Anselmo Lastra, Virtual Environments: A Survey of the Technology, University of North Carolina at Chapel Hill, TR93-033, Sept.1993
- [Hoppe93] Hugues Hoppe, Tony DeRose, Tom Duchamp, John McDonald, and werner Stuetzle. Mesh Optimizaition. In *SIGGRAPH '93* Proc., pp.19~26, Aug. 1993
- [Hoppe96] Hugues Hoppe, Progressive Meshes, In Holly Rushmeier, editor, *SIGGRAPH '96* Proceedings, New Orleans:ACM Press, 1996. 99~108.

- [Hoppe97] Hugues Hoppe. View-dependent refinement of progressive meshes. In *SIGGRAPH'97 Proc.*, pp. 189~198, August 1997.
- [Hoppe98] Hugues Hoppe. Smooth view-dependent level-of-detail control and its application to terrain rendering. In *IEEE Visualization '98 Conference Proceedings*, pp.35~42,516, Oct 1998.
- [Hudson97] T. Hudson, D. Manocha, J. Cohen, M. Lin, K. Ho_, and H. Zhang. Accelerated occlusion culling using shadow frusta. In *Proc. of ACM Symposium on Computational Geometry*, pp.1~10, 1997.
- [Hughes96] Merlin Hughes, Anselmo A. Lastra, and Edward Saxe. Simplification of global-illumination meshes. *Computer Graphics Forum*, 15(3):339~345, August 1996. *Proc. Eurographics '96*.
- [Isler96] Veysi Isler, Rynson W. H. Lau, Mark Green, Real-time Multi-resolution Modeling for Complex Virtual Environments, *VRST'96*, pp. 11~19, July, 1996.
- [Kalvin91] Calvin A. D., Cutting, C. B., Haddad, B. and Noz, M. E., Constructing topologically connected surfaces for the comprehensive analysis of 3D medical structures. *SPIE Vol. 1445 Image Processing*, 1991, 247~259.
- [Kalvin94] Alan D. Calvin and Russell H. Taylor. Superfaces: Polyhedral approximation with bounded error. In *Medical Imaging: Image Capture, Formatting, and Display*, volume 2164, pp.2~13. SPIE, February 1994.
- [Kalvin96] Calvin A. D. and Taylor, R. H., Superfaces: Poligonal mesh simplification with bounded error. *IEEE C.G. & A.*, 1996, 16(3), 64~77.
- [Kanada95] Kanada, T. , Narayanan, P. J. and Rander, P. W. Virtualized reality: Concepts and early results. In *IEEE Workshop on Representations of Visual Scenes*, pp. 69~76, Cambridge, MA, June 1995.
- [Kang96] Kang, S. B. and Szeliski, R 3-D scene data recovery using omnidirectional multibaseline stereo. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'96)*, pp. 364~370, San Francisco, CA, June 1996. IEEE Computer Society Press.
- [Klein96] Reinhard Klein, Gunther Liebich, and W. Straßer, Mesh reduction

- with error control. In *Proceedings of Visualization '96*, pp.311~318, October 1996.
- [Kobbelt98] Leif Kobbelt, Swen Campagna, and Hans Peter Seidel. A general framework for mesh decimation. In *Proc. Graphics Interface '98*, pp.43~50, June 1998.
- [Koch95] Koch, R. 3-D surface reconstruction from stereoscopic image sequenderink. In *Fifth International Conferende on Computer Vision (ICCV'95)*, pp. 109~114, MIT, Cambridge, MA, June 1995. IEEE Computer Society Press.
- [Kong90] X. Kong, H. Everett, G. Toussaint, The Graham Scan Triangulates Simple Polygon, *Pattern Recognition Letters* 11, Nov. 1990, pp. 713~716
- [Krueger91] Myron w. Krueger, *Artifitial Reality II*, Editors: Peter S. Gordon Helen M. Goldstein, Addison-Wesley Publishing Company, Inc., U.S.A. 1991
- [Kumar95a] Kumar, R. et al. Representation of scenes from collections of images. In *IEEE Workshop on Representations of Visual Scenes*, pp. 10~17, Cambridge, MA, June 1995.
- [Kumar95b] S. Kumar and D. Manocha. Efficient rendering of trimmed nurbs surfaces. *Computer-Aided Design*, pp. 509~521, 1995.
- [Kumar96] S. Kumar. *Interactive Rendering of Parametric Spline Surfaces*. PhD thesis, Department of Computer Science, University of N. Carolina at Chapel Hill, 1996.
- [Kumar97] S. Kumar and D. Manocha and H. Zhang and K. Hoff, Accelerated walkthrough of large spline models. In *Proc. of ACM Symposium on Interactive 3D Graphics*, pp.91~102, 1997.
- [Kumar99] Subodh Kumar, *Interactive Display of Spline Models*, SIGGRAPH'99 course #20, 1999
- [Lancaster86] Peter Lancaster and Kęstutis Salkauskas. *Curve and Surface Fitting: An Introduction*. Academic Press, London, 1986.
- [Lastra96] Anselmo A. Lastra, *Technology for Virtual Reality*, Siggraph'96 Course #14, 1996

- [Lastra97] Anselmo Lastra, Henry Fuchs, Programming Virtual worlds, Siggraph'97 Course #29, 1997
- [Lathrop99] Olin Lathrop, Virtual Reality, Siggraph'99 Course #18, 1999
- [Lau98] Rynson Lau, Mark Green, Danny To, and Janis Wong. Real-time continuous multi-resolution method for models of arbitrary topology. *Presence: Teleoperators and Virtual Environments*, 7(1):22~35, February 1998.
- [Laveau94] Laveau, S. and O. Faugeras, 3-D Scene Representation as a Collection of Images. In *12th International Conference on Pattern Recognition (ICPR'94)*, volume 1, pp.689~691, Jerusalem, Israel, October 1994. INRIA, Technical Report, No.2205, February 1994.
- [Levoy96] Levoy, M. and Hanrahan P. Light field rendering. In *Computer Graphics (SIGGRAPH'96)*, pp. 31~42, August 1996.
- [Li98a] 李洪举, “基于图象的建模和绘制研究”, 博士学位论文, 中国科学院软件研究所, 1998年12月.
- [Li98b] 李捷 唐泽圣, 三维复杂模型的实时连续多分辨率绘制, 计算机学报, 21(6):481~491, 1998
- [Lindstorm98] Peter Lindstrom and Greg Turk. Fast and memory efficient polygonal simplification. In *IEEE Visualization'98 Conference Proceedings*, pp.279~286,544, Oct 1998.
- [Liu98] 刘学慧 虚拟现实中三维复杂几何形体的层次细节模型的研究. 博士研究生学位论文,中科院软件所, 1998
- [Loop87] C. Loop, Smooth subdivision surfaces based on triangles, Master's thesis, University of Utah, Department of Mathematics, 1987
- [Loop94] C. Loop, Smooth spline surfaces over Irregular Meshes, *Computer Graphics Proceedings, Annual Conference Series*, 303~310, 1994
- [Lounsbery94] Michael Lounsbery, J. M. Multiresolution analysis for surfaces of arbitrary topological type. PhD thesis, Dept. of Computer Science and Engineering, U. of Washington, 1994
- [Lounsbery97] Michael Lounsbery, Tony D. DeRose, and Joe Warren.

- Multiresolution analysis for surfaces of arbitrary topological type. *ACM Trans. on Graphics*, 16(1):34~73, 1997.
- [Luebke95] D. Luebke and C. Georges. Portals and mirrors: Simple, fast evaluation of potentially visible sets. In *ACM Interactive 3D Graphics Conference*, Monterey, CA, 1995.
- [Luebke96] David Luebke. Hierarchical structures for dynamic polygonal simplification. TR 96-006, Department of Computer Science, University of North Carolina at Chapel Hill, 1996.
- [Luebke97] David Luebke and Carl Erikson. View-dependent simplification of arbitrary polygonal environments. In *SIGGRAPH'97 Proc.*, pp.199~208, August 1997.
- [Luken93] W.L. Luken. Tessellation of trimmed NURB surfaces. In *SIAM Conference on Geometric Design*, Tempe, AZ, 1993.
- [Low97] Kok-Lim Low and Tiow-Seng Tan. Model simplification using vertex-clustering. In *1997 Symposium on Interactive 3D Graphics*. ACM SIGGRAPH, 1997.
- [Ma98] 马小虎 潘志庚 石教英, 基于三角形移去准则的多面体模型简化算法, *计算机学报*, 21(6):492~498, 1998
- [Machover94a] Carl Machover, *MULTIMEDIA FUTURES*, Siggraph'94 Course #18, 1994
- [Machover94b] Carl Machover, *Fundamentals and Overview of Computer Graphics*, Siggraph'94 Course #18, 1994
- [Mallat89] Stephane G. Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 11(7):674~693, July 1989.
- [Manocha99] Dinesh Manocha, *Interactive Walkthroughs of Large Geometric Datasets*, *SIGGRAPH'99* course #20
- [Maruya95] Makoto Maruya. Generating a texture map from object-surface texture data. *Computer Graphics Forum*, 14(3):397~405, 506~507, 1995. *Proc. Eurographics '95*.
- [McMillan95] McMillan, L., G. Bishop, Plenoptic Modeling: An Image-Based Rendering System, *Computer Graphics (SIGGRAPH'95)*

- Conference Proceedings*), August 6-11, 1995, pp 39~46.
- [Melax98] Stan Melax. A simple, fast, and effective polygon reduction algorithm. *Game Developer*, pp. 44~49, November 1998.
- [Mine94] Mark R. Mine, Interaction in a Virtual Environment, Siggraph'94 Course #17, 1994
- [Mine95] Mark.R Mine, Virtual environment interaction techniques. University of North Carolina TR95-018, 1995
- [Mine96] Mark.R Mine, The Virtual World: Nice Place To Visit, But Would You Want To Work There? Siggraph'96 Course #31, 1996
- [Molnar92] S. Molnar, J. Eyles, and J. Poulton. Pixel ow: High speed rendering using image composition, *Proceedings of ACM Siggraph*, 26(2):231~248, 1992.
- [Nasri91] A. H. Nasri, Surface Interpolation on Irregular Networks with Normal Conditions, *Computer Aided Geometric Design* 8(1991), 89~96
- [Pan98] 潘志庚 马小虎 石教英, 多层次细节模型自动生成技术综述。 *中国图象图形学报*, 1998, 3 (9): 754~759
- [Pausch94] Randy Pausch, Software Development Environments and Architectures for Virtual Reality, Siggraph'94 Course #2, 1994
- [Peters95] J. Peters, C^1 Surface Splines, *SIAM J. Numer. Anal* 32, 2(1995), 645~666
- [Popović97] Jovan Popović and Hugues Hoppe. Progressive simplicial complexes. In *SIGGRAPH'97 Proc.*, pp. 217~224, 1997.
- [Prenter75] P. M. Prenter. *Splines and Variational Methods*. John Wiley & Sons, New York, 1975.
- [Preparata85] Franco P. Preparata and Michael I. Shamos. *Computational Geometry: an Introduction*. Springer-Verlag, New York, NY, 1985.
- [Reddy96] M. Reddy. SCROOGE: Perceptually-driven polygon reduction. *ComputerGraphics Forum*, 15(4):191~203, 1996.
- [Regan94] Regan M., Pose R., Priority Rendering with a Virtual Reality

- Address Recalculation Pipeline, *Computer Graphics (SIGGRAPH '94)*, pp.155~162, 1994.
- [Reif95] U. Reif, A unified approach to subdivision algorithms near extraordinary points, *Computer aided Geometric Design* 12 (1995): 153~174
- [Riedel98] Oliver Riedel, Ralf Breining, Ulrich Häfner, Roland Blach, Use of Immersive Projection Environments for Engineering Tasks, *Siggraph'98 Course #14*, 1998
- [Rippa92] Shmuel Rippa. Long and thin triangles can be good for linear interpolation. *SIAM J. Numer. Anal.*, 29(1):257~270, February 1992.
- [Robinett92] Warren Robinett, Richard. Holloway. Implementation of flying, scaling, and grabbing in virtual worlds, *Proc. 1992 Symposium on Interactive 3D Graphics*, Cambridge MA, March, 189~192.
- [Rockwood89] A. Rockwood, K. Heaton, and T. Davis. Real-time rendering of trimmed surfaces. In *Proceedings of ACM Siggraph*, pp. 107~17, 1989.
- [Ronfard96a] Rémi Ronfard and Jarek Rossignac. Full-range approximation of triangulated polyhedra. *Computer Graphics Forum*, 15(3), August 1996. *Proc. Eurographics '96*.
- [Ronfard96b] Rémi Ronfard and Jarek Rossignac. Full-range approximation of triangulated polyhedra. Technical Report IBM Research Report RC 20423, IBM T. J. Watson Research, Yorktown Heights, NY, 1996.
- [Rossignac93] Jarek Rossignac and Paul Borrel. Multi-resolution 3D approximations for rendering complex scenes. In B. Falcidieno and T. Kunii, editors, *Modeling in Computer Graphics: Methods and Applications*, pp.455~465, Berlin, 1993. Springer-Verlag. *Proc. of Conf., Genoa, Italy, June 1993*.
- [Schaufler95] G. Schaufler and W. Stürzlinger. Generating multiple levels of detail from polygonal geometry models. In M. Göbel, editor, *Virtual Environments '95 (Eurographics Workshop)*, pp.33~41. Springer Verlag, January 1995.
- [Salesin92] David Saesin, Filippo Tampieri, Grouping Nearly Coplanar Polygons into Coplanar Sets, David Kirk Ed. *Graphics Gems III*,

- Academic Press INC. 1992
- [Schmitt86] F. J. M. Schmitt, Barsky B. A., and Wen-Hui Du, An Adaptive Subdivision Method for Surface-Fitting from Sampled Data, *Computer Graphics*, 20(4):179~188, 1986
- [Schroeder92] William J. Schroeder, Jonathan A. Zarge, and William E. Lorensen. Decimation of triangle meshes. *Computer Graphics (SIGGRAPH '92 Proc.)*, 26(2):65~70, July 1992.
- [Schroeder97] William J. Schroeder. A topology modifying progressive decimation algorithm. In *IEEE Visualization 97 Conference Proceedings*, pp. 205~212,545, 1997.
- [Sedgewick83] Robert Sedgewick, *Algorithms*, Addison-Wesley, 1983
- [Seitz95] Seitz, S.M. and Dyer C. Physically-valid view synthesis by image interpolation. In *IEEE Workshop on Representations of Visual Scenes*, pp. 18~25, Cambridge, MA, June 1995.
- [Seitz96a] Seitz, S.M. and Dyer C. Toward image-based scene representation using view morphing. In *13th International Conference on Pattern Recognition (ICPR '96)*, volume 1, pages 84-89, Vienna, Austria, August 1996.
- [Seitz96b] Seitz, S.M. and Dyer C. View morphing: Synthesizing 3D metamorphoses using image transforms. In *Computer Graphics (SIGGRAPH' 96)*, pp.21~30, August 1996.
- [Shade96] Jonathan Shade, Dani Lischinski, David Salesin, Tony DeRose, and John Snyder. Hierarchical image caching for accelerated walkthroughs of complex environments. In Holly Rushmeier, editor, *SIGGRAPH 96 Conference Proceedings*, New Orleans:ACM Press, 1996. 75~82.
- [Shade98] Jonathan W. Shade, Steven J. Gortler, Li-wei He, and Richard Szeliski. *Layered depth images*. In Michael Cohen, editor, *SIGGRAPH 98 Conference Proceedings*, Addison Wesley:ACM Press, 1998. 231~242.
- [Shum99] Heung-Yeung Shum, Li-Wei He, Rendering with Concentric Mosaics, In Alyn Rockwood, editor, *SIGGRAPH 99 Conference Proceedings*, Addison Wesley: ACM Press, 1999. 299~306.
- [Soucy96a] Marc Soucy and Denis Laurendeau. Multiresolution surface

- modeling based on hierarchical triangulation. *Computer Vision and Image Understanding*, 63(1):1~14, 1996.
- [Soucy96b] Marc Soucy, Guy Godin, and Marc Rioux. A texture-mapping approach for the compression of colored 3D triangulations. *The Visual Computer*, 12(10):503~514, 1996.
- [Suter94] David Suter. Fast evaluation of splines using poisson formula. *International Journal of Scientific Computing and Modeling*, 1(1):70~87, 1994.
- [Sutherland65] Sutherland, Ivan., The ultimate display. *Information Processing 1965: Proceedings of IFIP Congress 65*. 506~508, 1965
- [Szeliski90] Richard Szeliski. Fast surface interpolation using hierarchical basis functions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(6):513~528, June 1990.
- [Szeliski94a] R. Szeliski and J. Coughlan, Hierarchical Spline-Based Image Registration, *Proc.IEEE Computer Soc. Conf. Computer Vision and Pattern Recognition (CVPR 94)*, IEEE CS Press, Los Alamitos, Calif., pp194-201, 1994.
- [Szeliski94b] Szeliski, R. Image mosaicing for tele-reality applications. In *IEEE Workshop on Applications of Computer Vision (WACV'94)*, pages 44-53, Sarasota, December 1994. IEEE Computer Society Press.
- [Szeliski96] Szeliski, R. Video mosaics for virtual environments. *IEEE Computer Graphics and Applications*, 16(2): 22-30, March 1996.
- [Taubin93] Gabriel Taubin. An improved algorithm for algebraic curve and surface fitting. In *Fourth International Conference on Computer Vision (ICCV '93)*, pages 658~665, Berlin, Germany, May 1993.
- [Teller91] S. Teller and C.H. Sequin. Visibility preprocessing for interactive walkthroughs. In *Proc. of ACM Siggraph*, pp.61~69, 1991.
- [Terzopoulos88] Demetri Terzopoulos. The computation of visible-surface representations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(4):417~438, July 1988.
- [Terzopoulos96] Demetri Terzopoulos, *Artificial Life for Graphics, Animation, Siggraph'96 Course #36*, 1996

- [Torborg96] Torborg J., Kajiya J., Talisman: Commodity Real-time 3D Graphics for the PC, *Computer Graphics (SIGGRAPH '96)*, pp.353~364,1996.
- [Turk92] Greg Turk. Re-Tiling Polygonal Surfaces. *Computer Graphics*, 1992, 26(2): 55~64
- [Turk99] Greg Turk, James F. O'Brien, Variational Implicit Surfaces, Tech Report GIT-GVU-99-15, Georgia Institute of Technology, May 1999
- [Upstill90] Upstill, S., *The Render Man Companion*. Addison-Wesley, 1990
- [Vance98] Judy M. Vance, Current Applications of Virtual Reality to Engineering Problems, *Siggraph'98 Course #14*, 1998
- [Watt98] Alan Watt, Fabio Policarpo, *The Computer Image*, Editor: Stephen Spencer, Addison Wesley Longman Limited, Edinburgh Gate, Harlow, Essex CM20 1JE, England,1998
- [Welch94] William Welch and Andrew Witkin. Free-form shape design using triangulated surfaces. *Computer Graphics Proceedings, Annual Conference Series (SIGGRAPH 94)*, pages 247~256, July 1994.
- [Werner95] Werner, T. et al. Rendering real-world objects using view interpolation. In *Fifth International Conference on Computer Vision (ICCV'95)*, pp.957~962, MIT, Cambridge, MA, June 1995. IEEE Computer Society Press.
- [Wloka94] Matthias M. Wloka, Lag in Multiprocessor VR, *Siggraph'94 Course #2*,1994
- [Wong97] Wong, T. T. Image-based Rendering with Controllable Illumination. *the 8th Eurographics Workshop on Rendering*, June 1997.
- [Xia96] Julie C. Xia and Amitabh Varshney. Dynamic view-dependent simplification for polygonal models. In *Proceedings of Visualization'96*, pp.327~334, October 1996.
- [Xu99] 徐丹, “基于图象的建模和绘制在小波空间中的研究与实现”, 博士学位论文, 浙江大学计算机系, 1999年6月.
- [Zhang97] H. Zhang, D. Manocha, T. Hudson, and K. Hoff. Visibility culling

-
- using hierarchical occlusion maps. Proc. of ACM SIGGRAPH'97, 1997.
- [Zhou96] 周晓云 刘慎权, 基于特征角准则的多面体模型简化, 计算机学报, Vol.19 Suppl.: 217~223, 1996
- [Zhou98] 周昆 潘志庚 石教英, 基于三角形折叠的网格简化算法, 计算机学报, 21(6):506~513, 1998
- [Zorin96a] Denis Zorin, Peter Schroder, Wim Sweldens, Interpolating Subdivision for Meshes with Arbitrary Topology, In Holly Rushmeier, ed. SIGGRAPH'96 Conference Proceedings, New Orleans:ACM Press, 1996. 189~192.
- [Zorin96b] Denis Zorin, Peter Schroder, Wim Sweldens, Interpolating Subdivision for Meshes of Arbitrary Topology, Tech. Rep. CS-TR-96-06, Caltech, Department of Computer Science, Caltech, 1996
- [Zorin99] Denis Zorin, Subdivision Zoo, Siggraph'99 Course#37

致谢

本论文是在我的导师彭群生教授的精心指导下完成的。彭老师深厚的学术功底、敏锐的学术洞察力、严谨的治学作风和精益求精的工作精神给我留下了深刻的印象，并将永远成为我学习的典范，他的谆谆教诲将使我受益终身。就是在彭老师的影响下，我的博士生学习生涯才成为真正在各方面提高自己的阶段。衷心感谢彭老师给予我的一切！

浙江大学 CAD&CG 国家重点实验室一流的研究环境、浓厚的学术气氛和团结协作的精神也是本论文顺利完成的关键。很多好的想法都是在与许多老师同学的讨论中形成的。特别是鲍虎军研究员，他思维开阔，对本论文提出了很多建设性的意见。同时，他严谨的学风和契而不舍的治学态度一直激励着我去克服工作中的种种困难。此外，在论文的研究过程中，郑文庭博士、刘新国博士、方向博士、陈为博士等各位师兄弟给了我很多无私的帮助。在此，对他们表示衷心感谢！

时光荏苒，转眼已经在求是园中生活四年。期间，陈莉博士、徐丹博士、杨旭波博士、王毅刚博士、余正生博士、郁维博士、尹康康硕士、张艳芬硕士以及 CAD 实验室的很多其他老师同学在研究、学习和生活上给予我很大的帮助和支持，给我的博士生学习生涯增添了很多快乐。我还要特别感谢吕思超老师、胡敏老师、王融青老师等对我的热心帮助。

我还要对西北工业大学的高光焘教授和蔡青教授表示感谢。虽然我已从西北工业大学毕业多年，但他们一直给予我真诚的关心和帮助。是他们的热情鼓励，使我始终保持高昂的斗志和永不服输的精神。

我敬爱的父母亲不懈的关怀，一直是我面对人生各种挑战最强大的动力。养育之恩，终身难忘！同时，感谢我的哥哥、姐姐和亲人们多年来对我的关心和支持！

四年的求是园生活，收获的不仅仅是学业，而且还有纯真的友谊和美好的回忆。我将把求是园里每一片回忆永远珍藏！

曹卫群

两千年二月于求是园

发表论文情况

1. 曹卫群, 鲍虎军, 彭群生, “基于高斯球的近似共面合并多层次细节模型”, 《软件学报》已录用。
2. 曹卫群, 彭群生, 包建中, “实用化玻璃模 CAD 系统的设计与实现”, 《机械科学与技术》, Vol.18, No.1, 1999. pp.160~163。
3. 曹卫群, 高光焘, “微机压延模 CAD 专家系统的工程图自动输出子系统”, 《机械科学与技术》, Vol.16, No.5, 1997. pp.928~930。
4. Zhengsheng Yu, Weiqun Cao, Lizhuang Ma, Qunsheng Peng, “A New Method for Implicit Algebraic Surfaces Intersection”, Proceedings of CAD&Graphics'97, pp.439~442, Dec. 1997, Shenzhen, China.
5. 曹卫群, 刘新国, 鲍虎军, 彭群生, “基于分割插值的连续多分辨率模型”, 已投出。