

## 摘 要

由曲面的二维灰度图像重建其三维形状已经成为计算机视觉领域的一个研究热点。对该问题进行了深入的研究,提出了一种从单幅图像出发,采用轮廓线进行旋转曲面重建的方法。该方法从一幅二维灰度图像中提取旋转体作为研究对象,通过一种自适应的 Harris 角点检测方法获取有效轮廓线及其分段曲线,根据分段曲线类型及其特征信息构造旋转体的对称轴,结合旋转体的几何特性实现了旋转曲面的三维重建。实验证明,该方法可以根据旋转体的二维灰度图像方便有效的重建出其三维形状。

本文的研究工作主要体现在以下几个方面:

1、利用数码相机获取的物体图像往往难以满足实验过程的要求,为了使重建结果尽量接近真实形状,必须对图像进行处理。主要是对图像进行平滑处理,去除噪声,然后进行必要的色彩转换以及阈值分割,以准确的从一幅二维图像中提取旋转体作为本文的研究对象;

2、介绍一种自适应的 Harris 角点检测方法,以便根据角点检测序列确定重建过程所需要有效轮廓线;重点讨论如何使用堆栈技术辅助八邻域搜索过程,以便对两条有效轮廓线分别进行轮廓跟踪,从而获取分段曲线及其特征信息;

3、由于本文主要以母线中含有圆弧段成分的旋转体作为研究对象,讨论如何根据几何性质设置判断条件来识别分段曲线的类型,进而重点探讨如何依照分段曲线的类型构造旋转体的对称轴;

4、相机拍摄的旋转体往往呈现非理想状态(非垂直摆放),为了处理的方便,设计了相应的坐标变换,将旋转体纠正为垂直摆放的状态,以便之后获取旋转体对象的三维信息;

5、充分利用旋转曲面的几何特性,结合有效轮廓线及对称轴信息,获取旋转体对象的三维信息;

6、针对本文理论成果进行了实践,以 OpenGL 和 VC++6.0 为开发工具实现了一个基于轮廓线的旋转曲面重建系统,并且为用户提供了良好的交互操作。

同时,为了体现完整性,在本文中对三维图形渲染的操作流程以及 OpenGL 的使用机制进行了简要的介绍。

**关键词:** 旋转曲面, 三维重建, 有效轮廓线, 对称轴, 角点检测, 特征点

## Abstract

3D reconstruction from the 2D grey image has been a research hot spot in the field of computer vision. Making an intensive study on the hot spot, and proposing a method which adopts the contour lines from single image to complete the reconstruction of the rotated curved surface. The method extracts the rotated object as the research target, adopts an adaptive method of Harris corner detection to acquire valid contour lines and their sub-curves, and then acquires the axis of rotated object according to the type of sub-curves and information of characteristic points, combine the geometric properties of rotated object to realize the 3D reconstruction of curved surface. The experiment indicates that the method could conveniently reconstruct the 3D shape of the rotated object according to its 2D grey image.

The paper focuses on the following aspects:

1、Using a digital camera to obtain image of object is often difficult to meet the requirements of the experiment, in order to enable the result of reconstruction to be the real shape as close as possible, the image should be processed. The processing mainly include image smoothing, noise removal, and the necessary color conversion and threshold segmentation, so that it can extract the rotated object as the research target from a 2D image.

2、An adaptive method of Harris corner detection has been introduced, so the valid contour lines required by reconstruction could be determined in accordance with the corners' sequence; Focus on the solution how to use stack technology to assist the eight neighborhood search process to trace the two valid contour lines, and then obtain the sub-curves and their characteristics points;

3、The paper mainly takes the rotated object whose generatrix contains circle arc as the research target, and discuss how to identify the type of the sub-curve according to the geometric characteristics, and then focus on the solution how to construct the axis of the rotated object in accordance with the type of the sub-curves;

4、The rotated object acquired by camera presents a non-ideal status(not vertical status), in order to deal with it conveniently, design the related coordinate transformation to correct the object's status to the vertical placement, so that the 3D information could be obtained.

5、Making full use of the geometric properties of rotated surface, combine the valid contour lines and axis to acquire the 3D information of the rotated object.

6、AS the practice of the theories work of this paper, use OpenGL and VC++ 6.0 to complete 3D reconstruction of rotated surface based on contour lines, and it also provides good interactive operations to users;

Meanwhile, in order to embody the integrity of the paper, it introduces the simple operations of 3D graphics rendering process and using mechanisms of OpenGL.

**Keywords:** rotated curved surface, 3D reconstruction, valid contour lines, axis, corner detection, characteristic points

## 独创性声明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含本人为获得江南大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

签 名： 钱苏斌 日 期： 2009.8.14

### 关于论文使用授权的说明

本学位论文作者完全了解江南大学有关保留、使用学位论文的规定：江南大学有权保留并向国家有关部门或机构送交论文的复印件和磁盘，允许论文被查阅和借阅，可以将学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文，并且本人电子文档的内容和纸质论文的内容相一致。

保密的学位论文在解密后也遵守此规定。

签 名： 钱苏斌 导师签名： 林意  
日 期： 2009.8.14

# 第一章 绪论

## 1.1 引言

随着虚拟现实技术、工业制造业、医疗辅助设备等各个领域的快速发展,对曲面重建技术应用的需求逐步增大。因此,如何快速的获得曲面模型成为计算机图形学及计算机视觉领域的研究人员长期以来努力的目标。

目前,曲面重建技术在当前国内各项领域发展较为迅速。譬如,在医疗方面,较为典型的是多层螺旋 CT 曲面重建在肋骨不全骨折中的应用,由于多层螺旋 CT 在扫描速度、范围、图像分辨率及后处理等方面功能较优,并且 CT 轴位扫描具有不受部位厚薄、体位及投照条件的限制等特点,所以可提供的影像信息更加丰富、准确<sup>[1]</sup>;在工业应用方面,基于数字摄影测量理论,利用投影器和 CCD 相机,对工业领域中的小型工业零件,如钣金件的三维曲面重建进行了深入的研究<sup>[2]</sup>;此外,还有目前应用较为广泛的真实感人脸模型的细分曲面重建,提出一种利用两张正交照片和细分曲面进行真实感三维人脸建模的方法,并实现了不同模型间的三维变形。为了构造个性化的人脸几何模型,将网格简化、自由曲面变形和细分结合起来,得到多个层次细节下的人脸模型,再经过纹理融合和映射,完成个性化的真实感三维人脸建模,同时利用线性插值实现了同拓扑真实感模型间的光滑变形<sup>[3]</sup>。实验结果表明:该方法不仅可以进行有效的真实感三维人脸建模,而且变形简单流畅,具有广阔的应用前景。综上,所有关于曲面重建的相关技术都在不同的领域促进了经济的发展,所以,对于这一科研应用领域的进一步探索是具有现实和长久意义的。

目前,在工业、制造业中广泛涉及的曲面重建可以分为如下两类:一类是自由曲面的重建,这类曲面需要较多的几何参数去精确的表示其形态,其常用的重建方法是三角 Bezier 曲面拟合或直接利用三角网格离散曲面<sup>[4]</sup>;另一类是特殊二次曲面的重建,譬如,常见的柱面,球面、锥面等旋转曲面。本文将立足于上述诸多领域中的旋转曲面的三维重建,进行深入的研究。

## 1.2 国内外研究的现状

### 1.2.1 三维重建技术综述

客观世界在空间上是三维的,而现有图像采集装置所获取的图像是二维的。尽管图像中含有某些形式的三维空间,但要真正在计算机中使用这些信息进行进一步的应用处理,就必须采用三维重建技术从二维图像中合理地提取并表达这些三维信息。三维重建技术能够从二维图像出发构造具有真实感的三维图形,为进一步的场景变化和组合运算奠定基础,从而促进图像和三维图形技术在航天、造船、考古、工业测量等领域的深入广泛的应用。

三维信息获取的技术手段多种多样,通常,人们获取物体三维模型的方式有三种,

下面将对这三种方法及其优缺点进行分析<sup>[5]</sup>。

第一种方法是利用建模软件构造三维模型。目前，在市场上比较知名的建模软件有3Dmax、Maya以及AutoCAD等等。它们的共同点在于利用一些基本的几何元素，通过一系列几何操作，如平移、旋转、拉伸以及布尔运算来构建复杂的几何场景。其优点是可以精确地构建出许多人造物体的三维模型，生成一些奇异的渲染效果。缺点是对这些软件的操作比较复杂，建模周期较长，同时需要熟练的操作人员，从而提高了制作成本。

第二种方法是通过仪器设备获取三维模型。如各种深度扫描仪(range scanner)等等。这些设备利用激光、超声波或者红外线测距，能够获得比较精确的三维数据，且使用比较方便，建模所需时间较少。但是其缺点在于设备一般比较昂贵，且携带不便。

第三种方法是利用图像或者视频来重建场景三维模型。图像是二维数据，但是在关于某一场景或物体的一幅或者多幅图像中可以找到许多线索，从中人们可以推知图像所记录的场景或者物体的几何信息。这些线索包括物体边与边之间的几何关系、两幅图像的视差关系、两幅图像中特征点的对应关系以及物体轮廓信息等等。这些线索是场景中物体所具有的，称为“被动线索”。有时候根据需要还可以创造线索，如在物体表面上用光线打上条纹或者制造出阴影，这样的人造线索称之为“主动线索”。对于使用主动线索的方法而言，可分为以下两类：一类是利用场景中已知形状的物体或者某些简单几何元素之间的关系进行建模；另一类是使用物体的轮廓信息。

被动法是在自然光照条件下获得三维信息的方法，其中包括：阴影恢复形状法、纹理恢复形状法、手动交互操作法；基于多幅图像的三维重建方法包括：立体视觉法<sup>[6,7]</sup>、运动图像序列法、光度立体学方法<sup>[5,8]</sup>等。

以上所述如图 1-1 来表示：

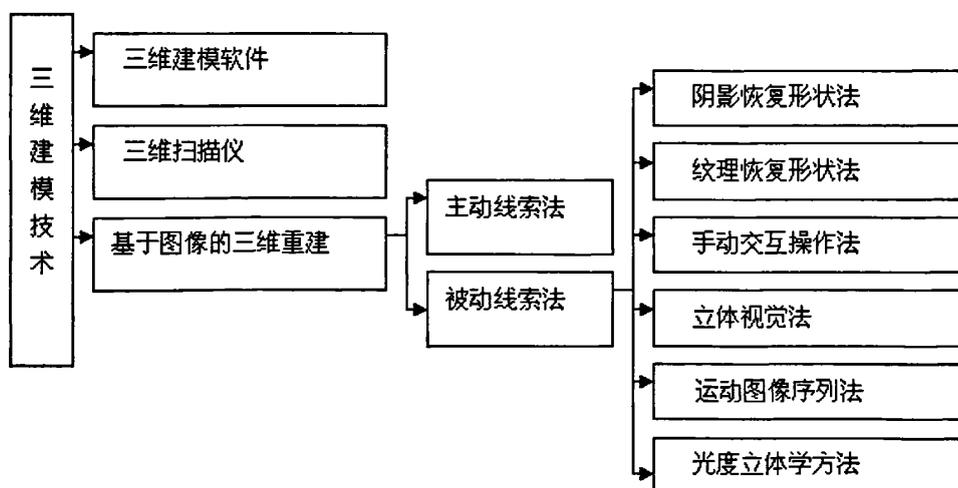


图 1-1 三维重建技术

Fig.1-1 3D reconstruction technology

## 1.2.2 基于图像的旋转曲面三维重建技术

现实生活中,许多物体的形态都表现为旋转体。譬如,在古文物中有许多陶瓷制品的形态都表现为旋转体;在工业测量中,许多工业零件都是旋转体或者由旋转体组成的。目前,在计算机视觉领域已经存在一些旋转体重建方法,如基于传统摄影测量技术中的中心、像点、物点三点共线原理进行的旋转体三维测量和重建方法<sup>[9,10]</sup>,但是由于旋转体及其影像间点与点之间的对应很难确定,所以这种视觉测量方法往往难以实现;文献[11][12]提出一些方法,但这些方法得到的只是旋转物体的相似模型,并没有能够实现旋转物体的绝对测量与重建;文献[13]提出了一种基于影像和物方重建的旋转体重建方法,首先利用旋转平台获取旋转体的影像,借助旋转平台的功能获取影像的参数,然后基于物方重建的方法计算旋转体的旋转轴,根据影像的参数和旋转体的旋转轴对影像进行纠正,利用纠正后的影像计算出旋转体的轮廓线,从而计算出整个旋转体的三维几何造型。该方法虽然在某些特殊领域具有一定的实用价值,但是由于影像参数计数精度不高,所以重建效果不是很理想。近年来,计算机视觉研究领域提出一种基于 Shape From Shading(SFS,基于阴影恢复形状)算法<sup>[14,15]</sup>进行曲面重建的方法。该方法主张首先从图像入手,采用图像处理技术对图像进行必要的处理,然后,根据图像的亮度实现旋转曲面的三维重建。上述方法不仅可以方便快捷的实现自由曲面的重建,避免了一般重建方法所要求的需要较多的几何参数去精确的表示曲面形态的缺陷,同时还可以实现包括旋转曲面在内的二次曲面的三维重建。然而,该方法仍然存在误差较大的不足。所以,根据旋转曲面所具备的特殊几何特性,结合上述方法重建速度快的特点,寻求出一种能够高速度、高精度的实现旋转曲面的三维重建的方法。

## 1.3 课题研究的意义

旋转体表面的三维重建技术在文物保护,陶瓷品制作以及工业测量等方面均具有重大的意义。譬如,在古文物中有许多陶瓷制品的形态都表现为旋转体。在工业测量中,许多工业零件都是旋转体或者由旋转体组成的。上述诸多情况都将涉及到旋转体表面(以下称为旋转曲面)的三维重建。因此,如何快速有效的获得旋转体的三维模型一直是计算机图形学及计算机视觉领域的研究人员努力的目标。

## 1.4 本论文的组织结构

本文的结构安排如下:

第一章:简要说明本课题的研究背景及研究意义,介绍了一些主流的三维建模技术以及国内外已有的由旋转体的二维灰度图像恢复其三维形状的重建方法,并分析了其应用价值与优缺点;

第二章:简要介绍三维重建过程中所涉及到的图像处理技术,对其中的轮廓跟踪算法与角点检测算法及其改进进行了介绍与分析;

第三章:简要介绍了三维图形真实感理论,并对其中的光照模型、材质进行分重点

的叙述；同时，简要地介绍了目前较为流行的三维图形软件库 OpenGL 及其在三维重建技术中的应用；

第四章：提出了一种基于轮廓线的旋转曲面三维重建方法，并对其具体实现进行了介绍与分析；

第五章：在本文的理论研究基础上，设计并实现了一个基于轮廓线的旋转曲面三维重建系统，使得用户能够对经过相机拍摄获得的旋转体的二维灰度图像进行相关处理，进而完成旋转曲面的三维形状的重新构建；

第六章：本文的结论和展望。

## 第二章 图像处理技术

在计算机科学领域,由二维灰度图像恢复物体的三维形状已日趋成为计算机视觉领域的研究热点之一。本文对该问题进行了深入的研究,充分利用旋转曲面的几何特性,提出了一种基于单幅图像,采用轮廓线进行旋转曲面重建的方法。但是,利用数码相机获取的物体图像往往会因为噪声等原因难以满足实验过程的要求,为了使重建结果尽量接近真实形状,必须对图像进行必要的预处理,如采用非线性平滑滤波器(G2)<sup>[16]</sup>,在保持图像细节的同时,对图像进行噪声消除。由于图像增强方面的处理较简单,这里即不做详细的介绍。下面即在本课题的研究背景下,对三维重建过程中所涉及的一些重要的图像处理算法进行介绍。

### 2.1 轮廓跟踪

#### 2.1.1 轮廓跟踪的基本原理

在识别图像中的目标时,往往需要对目标边缘作跟踪处理,也叫轮廓跟踪。顾名思义,轮廓跟踪就是通过顺序找出边缘点来跟踪边界的。在对旋转体对象经过阈值分割处理后,原图像成为二值图像。若图像是二值图像或图像中不同区域具有不同的象素值,并且每个区域内的象素值是相同的,则可以通过相应的轮廓跟踪算法得到图像分析所需要的轮廓线。

轮廓跟踪的基本方法<sup>[17]</sup>是:先根据某些严格的“探测标准”找出目标物体轮廓上的象素,再根据这些象素的某些特征用一定的“跟踪准则”找出目标物体上的其他象素。根据本文所研究课题的需要,首先介绍一种基于二值图像的轮廓跟踪算法。

#### 2.1.2 单目标区域轮廓跟踪算法

首先找到第一个边界象素的“探测准则”是:按照从左到右,从上到下的顺序搜索,找到的第一个黑点,一定是最左下方的边界点,记为 $A$ 。它的右、右上、上、左上四个邻点中至少有一个是边界点,记为 $B$ 。从 $B$ 开始找起,按照右、右上、上、左上、左、左下、下、右下的顺序找相邻中的边界点 $C$ 。如果 $C$ 就是 $A$ 点,则表明已经转了一圈,程序结束;否则从 $C$ 点开始继续找起,直到找到 $A$ 为止。判断是不是边界点很容易:如果它的上下左右四个邻点都不是黑点,则它即为边界点。这种算法要对每个边界象素周围的八个点进行判断,计算量比较大。下面介绍本文所引用的一种改进的单目标区域轮廓跟踪算法<sup>[18]</sup>。

#### 2.1.3 一种改进的单目标区域轮廓跟踪算法

首先按照 2.1.1 节所说的“探测标准”找到最左下方的边界点。以这个边界点起始,假设已经沿顺时针方向环绕整个图像一圈找到了所有的边界点。由于边界点是连续的,

所以每个边界点都可以用这个边界点对前一个边界点所张的角度来表示。因此可以使用下面的跟踪准则：从第一个边界点开始，定义初始的搜索方向为沿左上方；如果左上方的点是黑点，则为边界点，否则搜索方向顺时针旋转 45 度。这样一直找到第一个黑点为止。然后把这个黑点作为新的边界点，在当前搜索方向的基础上逆时针旋转 90 度，继续用同样的方法搜索下一个黑点，直到返回最初的边界点为止。如图 2-1 为这一轮廓跟踪算法的示意图。如图 2-2(a)所示，为 1 幅单目标区域的图像，该图像经过上述轮廓跟踪算法处理后，结果如图 2-2(b)所示。

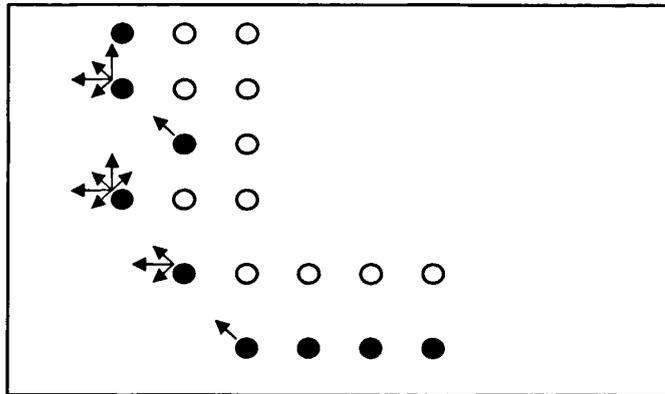
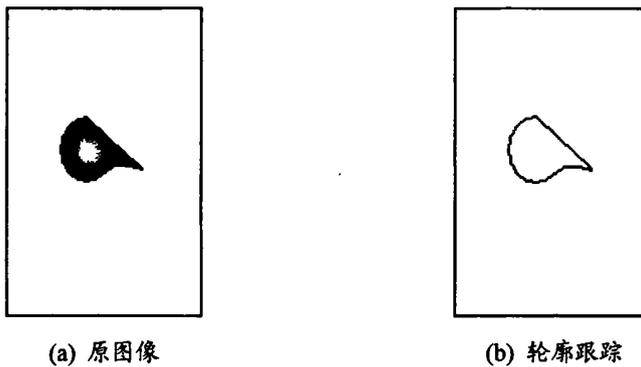


图 2-1 轮廓跟踪算法

Fig.2-1 Algorithm of contour tracing



(a) 原图像

(b) 轮廓跟踪

图 2-2 单图像多单目标轮廓跟踪

Fig.2-2 Single image, single object contour tracing

### 2.1.4 单图像多目标区域轮廓跟踪算法

对于一幅二值图像，既可以有一个目标区域，也可以同时存在多个目标区域，如图 2-3(a)所示，其中  $C1$ ,  $C2$ ,  $C3$  分别为三个相对独立的目标区域。当存在多个目标区域时，显然，2.1.3 节所述算法并不能有效的得到各目标区域的轮廓线。根据本文所研究课题的需要，设计出一种针对单图像多目标区域进行轮廓跟踪的方法。实验证明，该方法能够准确有效的从原图像中获取各目标区域的轮廓线，而且，随着目标区域的增长或者减少，算法性能并不会受到影响。

该算法采用堆栈技术辅助八邻域搜索过程，每次探测到一个目标区域的起始像素点

时, 首先将该起始像素点压入到堆栈中, 作为栈顶元素, 再将其弹出, 并对其进行八邻域搜索, 将搜索过程中满足相关条件的像素点压入堆栈中; 然后对堆栈中的每个像素点重复上述过程, 直至栈为空。此时, 对一个目标区域的轮廓跟踪工作结束。跟踪过程中, 根据目标区域个数, 设置多个一维点结构体数组, 分别用以存放跟踪出来的各个目标区域轮廓线上的像素点, 以便获得之后的分析工作所需要的特征信息。如图 2-3(b)所示, 即是利用上述方法对图 2-3(a)进行处理后的结果。其中,  $P_1$ ,  $P_2$ ,  $P_3$  分别为目标区域  $C_1$ ,  $C_2$ ,  $C_3$  的质心。

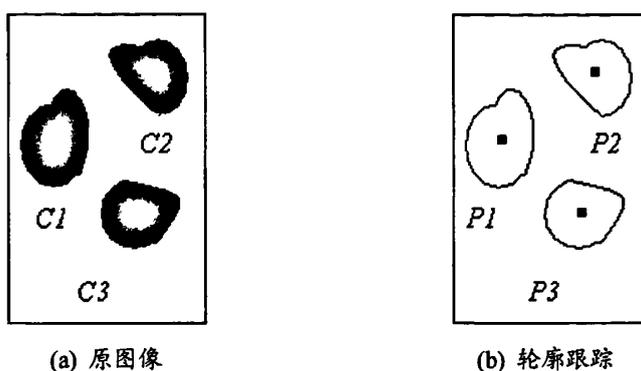


图 2-3 单图像多目标轮廓跟踪

Fig.2-3 Single image, multiple objects contour tracing

该算法可以有效获取各种目标区域的轮廓线(包括闭合和非闭合目标区域), 将被应用于本课题三维重建过程中非闭合轮廓线特征信息以及旋转轴的获取, 具体方法及应用将在第四章进行介绍。

## 2.2 角点检测

在计算机视觉中, 轮廓特征点的提取有着十分重要的意义。轮廓特征点主要包括角点、切点和拐点。角点相对于切点和拐点而言, 比较容易提取。此外, 角点在保留图像图形重要特征的同时, 也可以有效减少信息的数据量, 使其有效信息的含量很高, 从而极大地提高了计算的速度, 有利于图像的可靠匹配, 使得实时处理成为了可能。近年来, 角点检测在三维重建、运动估计、目标跟踪、目标识别、图像配准与匹配等计算机视觉领域起着非常重要的作用。

### 2.2.1 角点检测方法的分类

在计算机视觉领域, 角点被定义为一阶导数(即灰度的梯度)的局部最大所对应的点, 也就是说角点处不仅梯度的数值大, 而且梯度方向的变化速率也大, 所以角点提示了图像在两维空间内灰度变化剧烈的地方, 是和周围邻点有着明显差异的点。

目前, 常用的角点检测方法按照检测目标主要分为以下两类<sup>[19]</sup>: 第一类是基于轮廓曲线的角点检测方法, 第二类是基于灰度图像的角点检测方法。下面即对国内外与这两类检测方法相关的各种技术进行简要的介绍。

#### (1) 基于轮廓曲线的角点检测方法:

早在 1975 年, Rosenfeld A 等人<sup>[20]</sup>和 Freeman H 等人<sup>[21]</sup>就提出了通过计算角点强度  $k$  来提取角点, 不过这种方法虽然简单, 但容易受噪声干扰, 效果不是很理想。为了将干扰去除, 减少边缘毛刺干扰, Asada 等人<sup>[22]</sup>提出首先对边缘采用高斯平滑, 即减少了将局部弯曲度突然增大而误判为角点的概率。同时, 还可以通过计算曲线曲率的极值来确定轮廓曲线上的角点, 文献[23]提出了 B 样条来拟合曲线, 则角点即为 B 样条函数上曲率变化最大的点。此外, Witkin<sup>[24]</sup>和 Koenderink<sup>[25]</sup>提出基于尺度空间的图像分析理论后, 多尺度曲线分析成为解决该问题的主要方法, 在曲线尺度空间中, 随着曲线尺度由小变大, 一直保持较高弯曲度的点必定是所要求取的角点。

## (2) 基于灰度图像的角点检测:

基于灰度图像的角点检测方法可以分为两类。一类是基于梯度的方法<sup>[26]</sup>, 通过计算边缘的曲率来判断角点的存在性, 角点计算数值的大小不仅与边缘强度有关, 而且与边缘方向的变化率有关。另一类是基于模板的方法, 主要考虑像素邻域点的灰度变化, 即图像亮度的变化, 将与邻点亮度对比足够大的点定义为角点。

本文在利用角点检测方法时, 处理对象为一条经过阈值处理、轮廓跟踪等操作得到的闭合整体轮廓曲线。此时, 由于二值图像计算量不是很大, 可将该闭合整体轮廓线归类于二值图像的范畴, 从而保证了计算的实时性。同时, 对于二值图像而言, 基于灰度图像的各种角点检测方法对此仍然有效。根据本课题研究工作的实际需要, 从上述第二种分类中选择合适的方法对该闭合轮廓线进行角点检测。

### 2.2.2 Harris 角点检测方法的改进及应用

本文选择基于上述第二种分类方法的 Harris 角点检测算法对目标轮廓线进行角点提取。由于阈值是判断特征角点的唯一标准, 在对角点进行非极大值抑制时, 角点提取效果依赖于阈值的设定。阈值小会导致如图 2-4(a)所示的角点聚簇现象的发生, 阈值大又会出现如图 2-4(b)所示的丢失一部分角点信息。同时, 在用高斯函数对图像进行平滑处理的时候, 需要设定高斯方差和高斯模板大小。高斯方差和高斯模板大小不易确定。窗口太小, 在真实角点的邻域范围内易出现较多伪角点, 窗口太大, 又会使真实角点的位置发生较大幅度偏移。此外, 在求局部极大值时的邻域大小也将会影响提取角点的数目和容忍度。由此可见, 传统的 Harris 角点检测算法存在一定的局限。

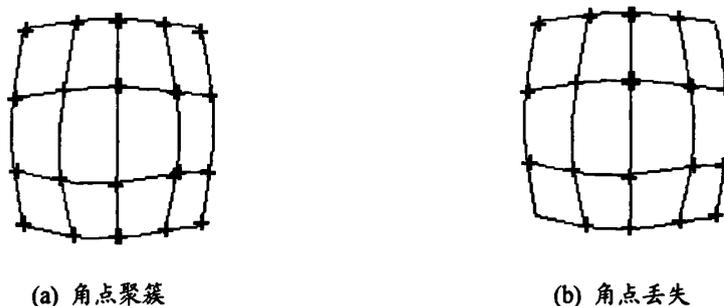


图 2-4 传统 Harris 角点检测

Fig.2-4 Traditional Harris corner detection

目前, 基于 Harris 算子已提出了较多的改进算法, 有亚像素级范围内的角点提取算法<sup>[27,28]</sup>, 基于小波变换的 Harris 多尺度角点检测算法<sup>[29]</sup>, 还有不同策略的自适应角点提取的算法: 文献[30]中采用阈值迭代策略, 文献[31]中采用固定块数分块法和邻近点剔除策略。此外, 还有根据棋盘格的几何特性提出了 SV 算子检测算法<sup>[32]</sup>, 以及基于 B 样条角点检测算法<sup>[33]</sup>。

通常情况下, Harris 检测需要根据图像的特性(如尺寸、纹理、灰度等), 对其所涉及到的阈值(*Threshold Value*)、局部极大值的邻域大小(*Width*)、高斯窗口的大小(*GaussWidth*)和高斯方差( $\sigma^2$ )四个参数进行手动调整。本文充分考虑了各个参数的相互作用和影响, 在此基础上引用了同研究室侯建辉同学的一种自适应的 Harris 棋盘格角点检测算法。

本文所研究处理的旋转体对象具有角点分布均匀, 个数确定的特点, 所以提取图像角点需要满足以下两个准则:

- (1) 每个角点的附近邻域内避免出现多个角点;
- (2) 角点数目和图像的长、宽有关, 据此得出角点区间;

当角点不满足上述准则时, 采取分步方法对参数进行调整。首先对阈值 *Threshold Value* 和非极大值抑制的邻域 *Width* 进行调整, 因为二者对角点的提取精度影响最为强烈。其次, 当仅仅通过这两个参数的调整仍不符合条件时, 对高斯函数宽度 *GaussWidth* 和高斯方差  $\sigma^2$  进行调整。通过分步调整参数思想, 可以不同程度地减少运算时间, 提高运算效率。

具体算法处理过程如下:

- (1) 对 *GaussWidth* 和  $\sigma^2$  赋最大初值;
- (2) 定义水平、垂直差分算子, 如图 2-5(a)、2-5(b)所示, 并利用其对图像进行滤波,

求得图像沿 X、Y 方向的灰度梯度  $I_x$ ,  $I_y$ , 同时求得  $I_x^2$ ,  $I_y^2$ ,  $I_{xy}$ ;

$$I_x^2 = I_x \times I_x, \quad I_y^2 = I_y \times I_y, \quad I_{xy} = I_x \times I_y \quad (2.1)$$

-1	0	1
-1	0	1
-1	0	1

(a) 水平差分算子

-1	-1	-1
0	0	0
1	1	1

(b) 垂直差分算子

图 2-5 差分算子

Fig.2-5 Difference operator

- (3) 对  $I_x^2$ ,  $I_y^2$ ,  $I_{xy}$  进行高斯滤波处理;
- (4) 计算角点量  $c(i, j)$ , 计算公式如 2.2 式所示;

$$c(i, j) = \frac{L^2(i, j) \times h^2(i, j) - Lx(i, j) \times Ly(i, j)}{L^2(i, j) + h^2(i, j) + 0.000001} \quad (2.2)$$

(5) 设置 *Threshold Value* 和 *Width*。如果角点不满足两条准则时, 用折半赋值的思想对 *Threshold Value* 和 *Width* 进行调整, 直到角点数目位于规定的区间范围内, 转(6); 否则, 转(1);

(6) 结束。

注:

(1) 各参数变化范围:

阈值: 5000-50000, 折半赋值法;

非极大值抑制的邻域宽度: 5-15, 步长为 1;

二维高斯窗口的宽度: 5-15, 步长为 1;

高斯函数的方差: 0.2-3.0, 步长为 0.2;

(2) 折半赋值法

当  $\min < \max$  时, 进入循环,  $Threshold\ Value \in (5000, 50000)$ ,  $\min = 5000$ ,  $\max = 50000$ ,

角点个数的区间为:  $CornerNum \in (NumMin, NumMax)$ ;

当  $CornerNum < NumMax$  时,  $mid = (\max + \min) / 2$ , 令  $\max = mid$ ; 反之,  $\min = mid$ ;

当  $\min > \max$  时, 结束循环。此时仍没满足准则, 调整其他参数。

如图 2-6(a)所示, 为某旋转体经过相关图像处理后的目标轮廓曲线, 对由该轮廓曲线围绕而成的目标区域使用上述自适应的 Harris 角点提取方法, 得到如图 2-6(b)所示的四个角点。

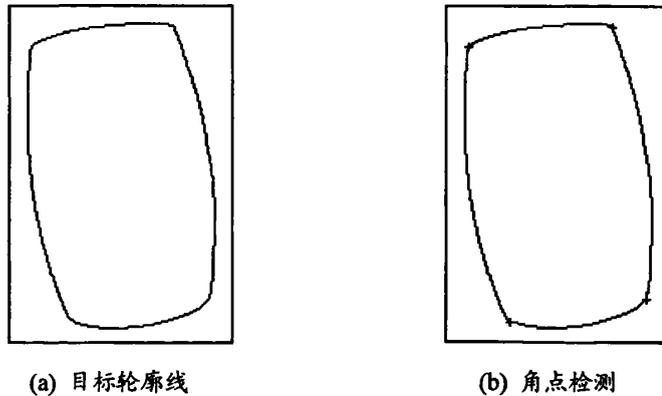


图 2-6 自适应的 Harris 角点提取

Fig.2-6 Adaptive Harris corner Detection

上述角点检测算法能够准确的检测出目标区域轮廓线上的角点, 有效的避免了传统 Harris 角点检测算法所存在的角点聚簇和角点丢失等问题, 将被应用于本课题三维重建过程所需要有效轮廓线的提取, 具体方法及应用将在第四章进行介绍。

## 第三章 真实感图形技术与OpenGL技术

### 3.1 计算机图形学

计算机图形学<sup>[34]</sup>是随着计算机及其外围设备的发展(特别是显示技术的发展)而产生的。自从上个世纪八十年代以来,计算机图形学得到了快速的发展,现已广泛应用于国民经济的各个领域。真实感图形技术一直是计算机图形学的一个重要研究内容,它对于进一步推动计算机图形学在计算机仿真、影视动画制作、虚拟现实、建筑和机械 CAD 等领域的应用发挥了极其重要的作用。本章首先将对真实感图形技术及其主流核心技术——真实感图形渲染技术进行简要的介绍。

### 3.2 真实感图形技术

多年来,真实感图形绘制技术一直是计算机图形学的主要研究内容之一。真实感图形在虚拟现实、几何造型、科学计算可视化等诸多领域得到了广泛的应用,它对人们的日常工作、学习和生活产生了深远的影响。下面将对真实感图形的主流技术做简要的介绍。

### 3.3 光照明模型

光源和材质,均属光照模型范畴,某种材质模型和与之匹配的某种光源模型相统一即是光照模型。为了产生明暗过渡自然的真实感图形,首先必须建立光照明模型(muminatinnModel 或 LightingMedel),它描述了物体表面的颜色与其空间位置、朝向、物理属性(反射率、折射率)关系。依据光照明模型、绘制算法具体计算对应屏幕上各像素的物体表面的颜色,并最终显示图形。光照明模型主要包括简单光照模型和整体光照模型,由于本文的应用需要,这里仅对简单光照模型进行介绍。

#### 3.3.1 简单光照明模型

当光照射在某一表面上时,它可能被吸收、反射或折射。其中入射到表面上的一部分光能被吸收并转化为热,其余部分被反射或折射。正是反射或折射部分的光使物体可见。光能中被吸收、反射或透射的数量决定于光的波长。若入射光中所有的光均被吸收,物体将呈黑色。若其中只有一小部分被吸收,则物体呈现“白色”。若某些波长的光被有选择的予以吸收,则离开物体表面的反射光和透射光将具有不同的能量分布。这时物体呈现出颜色。物体的颜色决定于它所选择吸收的那部分光的波长<sup>[35]</sup>。

##### 1、环境光

在避光的地方(如没有开灯的房间内),景物没有受到光源(如白炽灯、太阳)的直接照射,但其表面仍然具有一定的亮度使它们可见。这是因为光线在场景中经过复杂的传

播之后,形成了弥漫于整个空间的光线,称为环境光(ambient light 或 background light)。

环境光在空间中近似于均匀分布,即在任何位置、任何地方上强度都一样,记其亮度为  $I_a$ 。在分布均匀的环境光的照射下,物体表面呈现的亮度未必相同,因为它们具有不同的环境光反射系数(ambient reflection coefficient),记为  $K_a$ ,  $K_a$  较大者,看起来亮,  $K_a$  较小者,看起来暗。由此得到只包含环境光的光照明方程(描述光照明模型的方程)为:

$$I_e = K_a \cdot I_a \quad (3.1)$$

其中,  $I_e$  为物体表面所呈现的亮度。由这个方程可以看出,若一个物体的  $K_a=0$ ,则无论环境光多么强,该物体都是黑色的。并且,当同一物体表面的  $K_a$  为常数,则其上点的亮度相同。

## 2、漫反射

采用方程(3.1)绘制物体,虽然不同的物体具有不同的亮度,但同一物体的表面亮度是一个恒定的值,没有明暗的自然过渡,这是环境光照射产生的结果。因此,我们考虑用点光源照射物体,记点光源亮度为  $I_p$ 。在点光源的照射下,物体表面的不同部分亮度不同。亮度的大小依赖于它的朝向以及它与点光源之间的距离。一个粗糙的、无光泽的物体(如粉笔)表面对光的反射表现为漫反射(diffuse reflection)。在表面某点上,对入射光在各个方向上呈强度相同的反射。因而无论从哪个角度看,该点的亮度都是相同的。

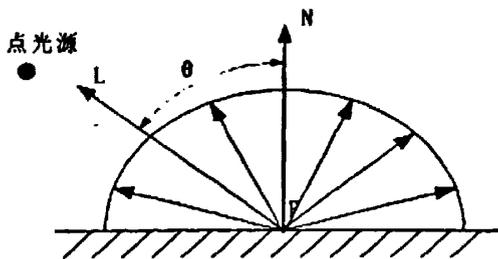


图 3-1 漫反射光强度与入射角  $\theta$  有关

Fig.3-1 Intensity of diffuse reflection light being related with Incidence angle  $\theta$

对于给定的表面,记其在  $P$  点的法向量为  $\vec{N}$ ,从  $P$  点到点光源的矢量为  $\vec{L}$ ,  $\theta$  为  $\vec{N}$  与  $\vec{L}$  的夹角,  $P$  点的漫反射光亮度  $I_d$  由朗伯(Lambert)余弦定率确定:

$$I_d = I_p K_d \cos \theta, \quad \theta \in (0, \frac{\pi}{2}) \quad (3.2)$$

其中,  $K_d \in [0,1]$ ,为物体表面的漫反射系数(diffuse reflection coefficient)。由上式可以看出,  $I_d$  只与入射角  $\theta$  有关,而与反射角无关,也即与视点的位置无关。如图 3-2 所

示,  $I_d$  的强度在各个反射方向上相等。入射角  $\theta$  必须在  $0$  到  $\frac{\pi}{2}$  之间变化, 当  $\theta > \frac{\pi}{2}$  时反射光线将被物体自身遮挡而照射不到  $P$  点(对于不透明物体)。若  $\vec{N}$  与  $\vec{L}$  都已规范化为单位矢量, 则  $\cos\theta = \vec{L} \cdot \vec{N}$ , 则式(3.2)可以写成:

$$I_d = I_p \cdot K_d (\vec{L} \cdot \vec{N}) \tag{3.3}$$

### 3、镜面反射与 Phong 模型

光滑的物体表面(如金属或塑料表面)在点光源的照射下会形成一块特别亮的区域, 即所谓的“高光区”, 它是由物体表面对入射光进行镜面反射引起的。镜面反射遵循反射定律, 反射光与入射光位于表面法向两侧。对于理想反射面(如镜面), 入射角等于反射角, 观察者只能在反射方向上才能看到反射光, 而偏离了该方向则看不到任何反射光。对于非理想反射面, 如苹果、塑料球等的表面, 镜面反射情况由 Phong 模型给出。

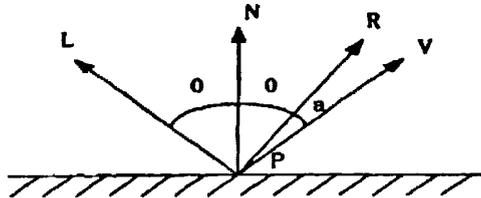


图 3-2 非理想镜面反射

Fig.3-2 Non-ideal regular reflection

如图 3-2 所示,  $P$  为物体表面上一点,  $\vec{L}$  为从  $P$  指向光源的矢量,  $\theta$  为入射角(反射角),  $\vec{V}$  为从  $P$  指向视点的矢量,  $\alpha$  为  $\vec{V}$  与  $\vec{R}$  的夹角, Phong 模型表示为:

$$I_s = I_p K_s \cos \alpha^n \tag{3.4}$$

其中,  $I_s$  为从  $\vec{v}$  方向上观察到的镜面反射光的亮度, 显然, 在  $\alpha = 0$  的方向上(即  $\vec{R}$  方向)  $I_s$  取得最大值, 则  $\cos \alpha = \vec{V} \cdot \vec{R}$  带入式(3.4), 得到:

$$I_s = I_p K_s (\vec{V} \cdot \vec{R}) \tag{3.5}$$

将镜面反射光加入方程(3.5), 得到一个简单照明模型:

$$I = I_e + I_d + I_s = K_a I_e + I_p [K_d (\vec{L} \cdot \vec{N}) + K_s (\vec{V} \cdot \vec{R})^n] \tag{3.6}$$

该简单照明方程中包含了环境光、漫反射光与镜面反射光。

## 3.4 OpenGL 技术

OpenGL 即开放性图形库(Open Graphic Library)是 Silicon Graphics(SGI)公司开发的

三维工具软件包，它是一种可独立于操作系统和硬件环境的三维图形库<sup>[36]</sup>。OpenGL 本身具有超强大的图形绘制功能，包括绘制物体、启动光照、管理位图、纹理映射、动画、图像增强以及交互技术等功能<sup>[37]</sup>。综合以上功能，作为图形硬件的软件接口，OpenGL 主要是将三维的物体投影到一个二维的平面上，之后处理得到像素，进行显示。

### 3.4.1 OpenGL的功能与特点

OpenGL 是一个开放的三维图形软件包，它独立于窗口系统和操作系统。以 OpenGL 为基础开发的应用程序可以十分方便地在各种平台间移植。OpenGL 可以与 Visual C++ 紧密接口，便于实现复杂的有关计算和图形算法，可以保证算法的正确性和可靠性。下面对本文所涉及的一部分 OpenGL 操作作简要介绍：

#### 1、绘制

真实世界的各个物体都可以在计算机中用简单的点、线、面来描述。OpenGL 提供了丰富的图形绘制命令，可以方便绘制物体。OpenGL 图形库除了提供基本的点、线和多边形的绘制函数外，还提供了较复杂的三维物体(球、锥、多面体、茶壶等)以及复杂曲线曲面的绘制函数。

#### 2、变换

OpenGL 图形库的变换包括基本变换和投影变换。基本变换有平移、旋转、变比和镜像四种变换。投影变换有平行投影(又称正射投影)和透视投影两种变换。下面将对本课题研究背景下所涉及到的平行投影作简要介绍。

平行投影又叫正射投影，这种投影的视景体是一个矩形的平行管道，也就是一个长方体，如图 3-3 所示。正射投影的最大一个特点是无论物体距离相机多远，投影后的物体大小尺寸不变。这种投影通常用在建筑蓝图绘制和计算机辅助设计等方面，这些行业要求投影后的物体尺寸及相互间的角度不变，以便施工或制造时物体比例大小正确。

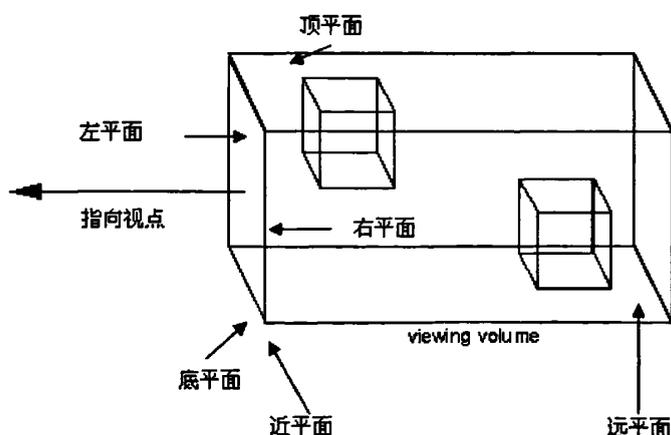


图 3-3 平行投影视景体

Fig.3-3 Orthographic projection view

#### 3、颜色模式设置

OpenGL 中颜色模式有两种,即 RGBA 模式和颜色索引(Color Index)模式。在 RGBA 模式下,所有的颜色定义全部用 R、G、B 三个值表示,有时还会加上 Alpha 值与透明度有关,即 RGBA 模式。在颜色索引模式下,像素的颜色用颜色索引值表示,该索引值指向颜色查找表(Lookup Table)中相应的 R、G、B 值。

#### 4、光照和材质设置

光照是增强物体真实感最重要的手段。对于大多数物体,只有对其加入光照效果之后,才会由鲜明的三维效果。而物体三维显示效果的不同,跟光源位置、光照颜色、物体材质的设置有很大关系。在简单光照模型中的光分为:辐射光(Emitted Light)、环境光(Ambient Light)、漫射光(Diffuse Light)、镜面光(Specular Light)<sup>[38]</sup>;材质用光反射率来表示;场景中物体最终反映到人眼的颜色是光的红绿蓝分量与材质红绿蓝分量的反射率相乘后形成的颜色。

### 3.4.2 OpenGL 的工作原理

OpenGL 能够绘制的图元包括点、线段和多边形,使用时可以在这几种图元模式之间选择。各个图元模式可以是相对立的,模式的选择、图元的定义以及其它操作都必须通过调用相应函数来实现。

图元被定义为一组顶点,顶点构成线或多边形等基本元素。与顶点相关联的数据包括其坐标、颜色、法向量、纹理坐标和边标识等,在通常情况下 OpenGL 以相同的方法独立地顺序地处理每个顶点及其关联数据。下面将对本课题研究背景下所涉及到的顶点法向量进行介绍与分析。

法向量是几何图元的重要属性之一。物体的法向量定义了其表面的朝向,具体的说,是相对于光源的朝向,OpenGL 用这些向量来确定物体定点将接受多少光。

OpenGL 法向量定义函数为:

```
void glNormal3f(TYPE nx,TYPE ny,TYPE nz);
```

```
void glNormal3f(TYPE *v);
```

非向量形式定义法向量采用第一种方式,即在函数中分别给出法向三个分量值  $nx$ 、 $ny$  和  $nz$ ;向量形式定义采用第二种,即将  $v$  设置为一个指向拥有三个元素的指针,例如  $v[3]=\{nx,ny,nz\}$ 。

OpenGL 本身没有提供计算法向量的函数(计算法向量的任务由程序员自己去完成),但它提供了赋予当前顶点法向量的函数。一般,程序员需要完成以下两种法向量的计算:

(1) 平面法向的计算:在一个平面内,有两条相交的线段,假设其中一条为矢量  $\vec{w}$ ,另一条为矢量  $\vec{v}$ ,平面法向为  $\vec{N}$ ,则平面法向就等于两个矢量的叉积(遵循右手定则),即为  $\vec{N} = \vec{w} \cdot \vec{v}$ ;

(2) 曲面法向量的计算:对于曲面各顶点的法向计算有很多种,如根据函数表达式求偏导的方法等。但是在大多数情况,OpenGL 中的多边形并不是由曲面方程建立起来

的，而是由模型数组构成，这时候求取法向量的办法是将曲面细分成多个小多边形，然后选取小多边形上相邻的三个点  $v_1$ 、 $v_2$ 、 $v_3$  (当然三个点不能在同一直线上)，按照平面法向量的求取方法即可。

OpenGL 的工作原理<sup>[39]</sup>和显示原理如下图所示：

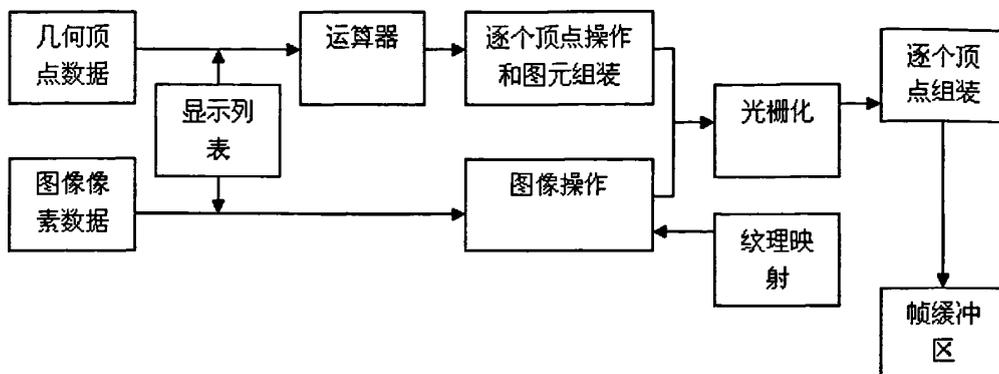


图 3-4 OpenGL 工作原理

Fig.3-4 OpenGL working principle

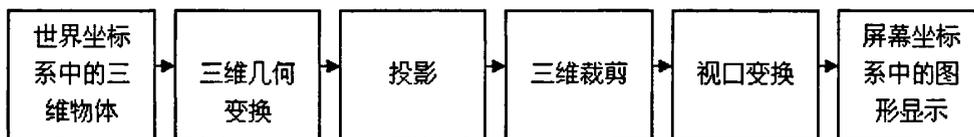


图 3-5 OpenGL 显示原理

Fig.3-5 OpenGL rendering principle

## 第四章 基于轮廓线的旋转曲面三维重建方法

### 4.1 概述

旋转体表面的三维重建技术在文物保护、器皿制作以及工业测量等方面均具有重大的意义。譬如,古文物中有许多陶瓷制品的形态都表现为旋转体;在工业测量中,许多工业零件都是旋转体或者由旋转体组成的。上述诸多情况都涉及到旋转体表面(以下称之为旋转曲面)的三维重建。因此,如何快速有效地获得旋转体的三维模型一直是计算机图形学及计算机视觉领域的研究人员努力地目标。

本章将提出一种基于轮廓线的旋转曲面三维重建方法,该方法充分利用旋转曲面的特殊几何性质,根据旋转体的二维灰度图像方便有效的重建出其三维形状。下面将对该方法的具体实现进行详细的介绍。

### 4.2 轮廓线设计

本文所研究的是旋转曲面的三维重建,旋转曲面往往可以通过将一条平面曲线  $C$ (又称之为母线)围绕一条对称轴  $L$  旋转形成,所以在重建过程中需要确定构成旋转曲面的这两大因素,一是平面曲线  $C$ ;二是对称轴  $L$ 。那么,首先确定曲线  $C$ 。这条平面曲线即旋转体的有效轮廓线,下面将对这条有效轮廓线的获取方法进行介绍。

#### 4.2.1 单目标轮廓线跟踪(整体轮廓线提取)

如图 4-1(a)所示,为 1 幅由 Sony707 数码照相机拍摄得到的花瓶,对其进行非线性平滑滤波处理及必要的颜色转换后获取的原图像。其中目标区域为一只花瓶,其瓶体表面为旋转曲面。对该灰度图进行阈值分割操作,结果如图 4-1(b)所示。根据重建过程的要求,首先需提取出该旋转瓶体的整体轮廓线。观察该位图,其中目标区域只有一个,所以利用 2.1.3 节中经过改进的单目标区域轮廓跟踪方法,得到如图 4-1(c)所示的整体轮廓线。

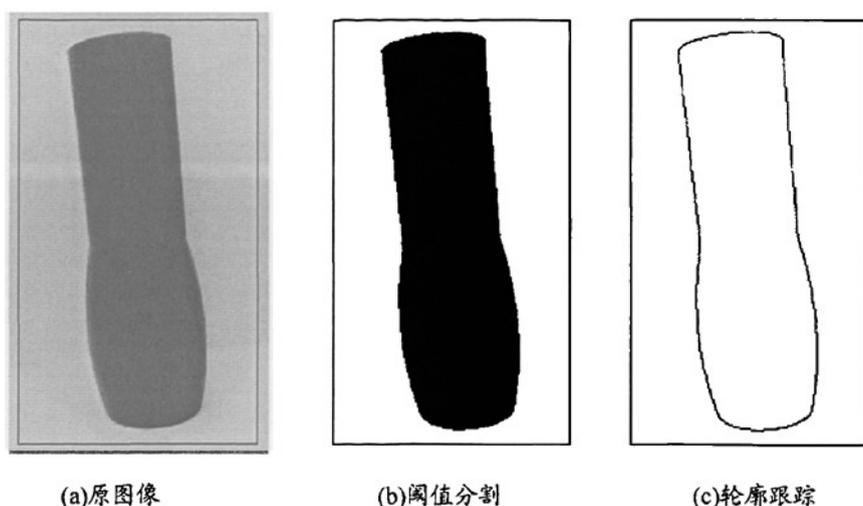


图 4-1 单轮廓线跟踪

Fig.4-1 Single contour tracing

#### 4.2.2 有效轮廓线提取

观察图 4-1(c)所示的旋转杯体的整体轮廓线,由上、下、左、右四部分组成。根据本文研究的需要,旋转曲面的构成仅需其中的左、右两条轮廓线(即母线),则上、下两条轮廓线为无效轮廓线,这里便涉及到如何将无效轮廓线截去,从而提取出有效轮廓线。

在数字图像处理过程中,物体所占区域边界的角点是一种重要的局部特征。到目前为止,关于角点的定义,在不同的角点检测方法中有不同的定义,主要有以下几种:

- (1) 角点是一阶导数(即灰度的梯度)的局部最大所对应的点;
- (2) 角点是两条及两条以上边缘的交点;
- (3) 角点指示了物体边缘变化不连续的方向;
- (4) 角点处的一阶导数最大,二阶导数为零;

(5) 角点处不仅梯度的数值大,而且梯度方向的变化率也大。意即,角点提示了图像在两维空间内灰度变化剧烈的地方,是和周围邻点有着明显差异的点。

根据上述定义,旋转杯体的整体轮廓线中必定存在符合上述定义之一的角点。观察图 4-1(c),上、下、左、右四条轮廓线的交点完全具备角点的特征,即可作为旋转瓶体所占区域边界的角点。

在此,初始化点结构体如下:

```
typedef struct tagPOINT2D
{
    long x;
    long y;
}POINT2D;
```

利用 2.2.2 节中所引用的一种自适应的 Harris 角点检测算法对 4-1(c)所示的整体轮廓

线进行角点检测,即可得到角点 $p_1, p_2, p_3, p_4, p_5, p_6$ ,结果如图4-2(a)所示,其中(角点用“+”标记);同时设置一维点结构体数组 $jpoint[]$ 存放各角点相应的坐标信息。那么,曲线 $\widehat{p_1 p_3 p_5}$ 和曲线 $\widehat{p_2 p_4 p_6}$ 即为重建所需要的有效轮廓线,而曲线 $\widehat{p_1 p_2}$ 和曲线 $\widehat{p_3 p_6}$ 则作为无效轮廓线,应当截去。

采用域截取方法,并结合一维点结构体数组 $jpoint[]$ 中所存放的各角点相应的坐标信息,将曲线 $\widehat{p_1 p_2}$ 和曲线 $\widehat{p_3 p_6}$ 截去,结果如图4-2(b)所示。

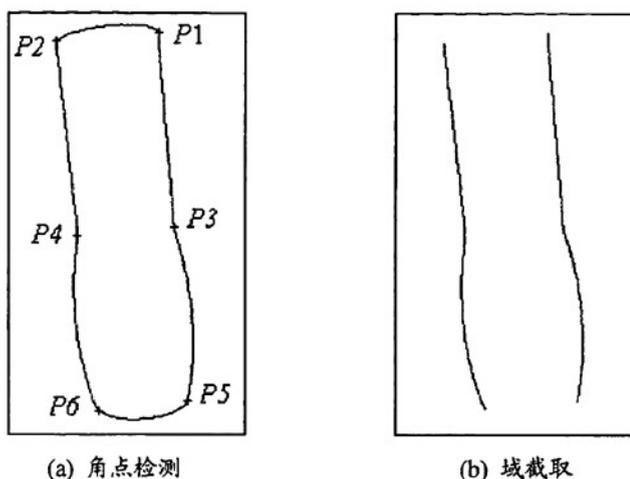
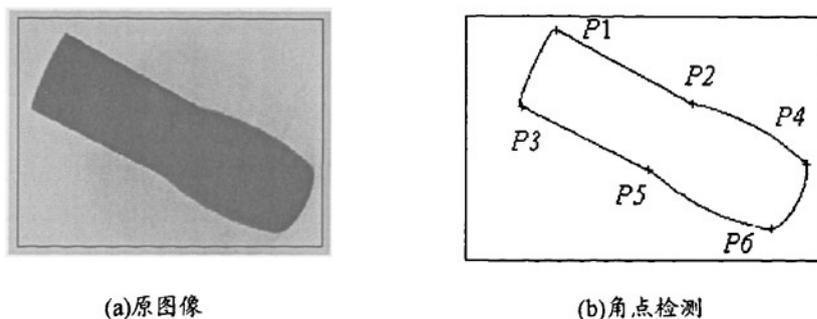


图 4-2 有效轮廓线提取

Fig.4-2 Validated contour extracting

在现实情况中,往往会因为相机拍摄角度或者物体摆放位置的不同,导致拍摄出来的物体呈现多种不同的状态。由于瓶体所呈现摆放状态的不同,其角点检测序列也会随之变化,从而有效轮廓线的提取也将会发生改变。图4-3(a)、4-3(c)所示的旋转瓶体,是通过改变相机拍摄角度或瓶体的摆放位置,使得同一瓶体呈现的两种不同的状态,对其分别进行上述一系列的图像处理操作,得到各自相应的角点分布,结果如图4-3(b)、4-3(d)所示。



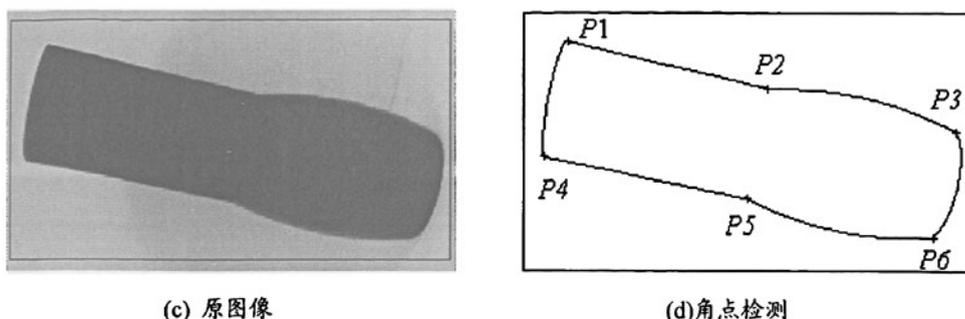


图 4-3 多种状态下角点分布

Fig.4-3 Corner distribution in multiple status

假设一维点结构体数组  $jpoint[]$  中依次存放了检测出来的六个角点。综上分析可得, 旋转体的角点检测序列会随着拍摄的瓶体所呈现的倾斜角度的变化而变化, 所以有效轮廓线的角点标记也将随着倾斜角度的变化而变化。以如图 4-2(a) 所示瓶体为例, 相对于垂直放置而言, 按照逆时针方向倾斜, 角度在  $0^\circ$  至  $90^\circ$  之间时, 大致可分为以下几种:

- (1) 如果同时满足条件:  $jpoint[2].x > jpoint[3].x$ ,  $jpoint[4].x > jpoint[5].x$ ,  
 $jpoint[1].y = jpoint[2].y$ ,  $jpoint[3].y = jpoint[4].y$ ,  $jpoint[5].y = jpoint[6].y$ ,

有效轮廓线:  $\overline{P_1P_3P_5}$ ,  $\overline{P_2P_4P_6}$ ;

- (2) 如果同时满足条件:  $jpoint[2].x < jpoint[3].x$ ,  $jpoint[4].x < jpoint[5].x$ ,

有效轮廓线:  $\overline{P_1P_3P_5}$ ,  $\overline{P_2P_4P_6}$ ;

- (3) 如果同时满足条件:  $jpoint[2].x > jpoint[3].x$ ,  $jpoint[4].x > jpoint[5].x$ ,

$jpoint[1].y \neq jpoint[2].y$ ,  $jpoint[3].y \neq jpoint[4].y$ ,  $jpoint[5].y \neq jpoint[6].y$ ,

有效轮廓线:  $\overline{P_1P_2P_4}$ ,  $\overline{P_3P_5P_6}$ ;

- (4) 如果同时满足条件:  $jpoint[2].x > jpoint[4].x$ ,  $jpoint[3].x > jpoint[5].x$ ,

有效轮廓线:  $\overline{P_1P_2P_3}$ ,  $\overline{P_4P_5P_6}$ ;

- (5) 如果同时满足条件:  $jpoint[2].x > jpoint[3].x$ ,  $jpoint[4].x < jpoint[5].x$ ,

有效轮廓线:  $\overline{P_1P_2P_5}$ ,  $\overline{P_3P_4P_6}$ ;

如上所示的旋转体对象表面为复杂的旋转曲面, 相对而言, 单一旋转曲面(圆柱、圆锥、圆弧等)的角点分布则较简单。假设现有一只表面为单一圆弧面的茶杯, 通过改变相机拍摄角度或瓶体的摆放位置, 使得同一杯体呈现两种不同的状态, 如图 4-4(a)、4-4(c) 所示, 对其分别进行上述一系列的图像处理操作, 得到各自相应的角点分布, 结果如图 4-4(b)、4-4(d) 所示。

假设一维点结构体数组  $jpoint[]$  中依次存放了检测出的四个角点, 那么旋转杯体的角点检测序列会随着拍摄的瓶体所呈现的倾斜角度的变化而变化, 所以其有效轮廓线也将随着倾斜角度的变化而变化。以如图 4-4(a) 所示杯体为例, 相对于垂直放置而言, 按照逆时针方向倾斜, 角度在  $0^\circ$  至  $90^\circ$  之间时, 大致可分为以下几种:

(1) 如果同时满足条件:  $jpoint[1].x > jpoint[2].x$ ,  $jpoint[3].x > jpoint[4].x$ ,

有效轮廓线:  $\widehat{P_1P_3}$ ,  $\widehat{P_2P_4}$ ;

(2) 如果同时满足条件:  $jpoint[1].x < jpoint[2].x$ ,  $jpoint[3].x < jpoint[4].x$ ,

有效轮廓线:  $\widehat{P_1P_2}$ ,  $\widehat{P_3P_4}$ ;

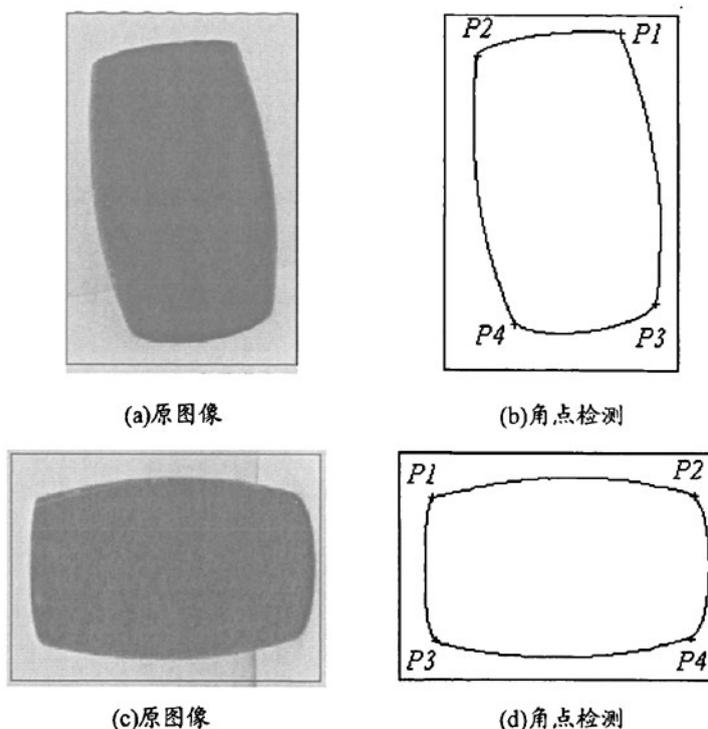


图 4-4 简单旋转曲面角点分布

Fig.4-4 Corner distribution in simple status

#### 4.2.3 有效轮廓线特征信息获取(分段曲线及其特征信息)

利用 2.1.4 中所介绍的多目标轮廓跟踪算法, 对两条有效轮廓线分别进行轮廓跟踪(双轮廓跟踪方法), 以获取其特征信息。由于复杂旋转曲面的有效轮廓线往往不是单一的直线或二次曲线, 而是由直线和二次曲线, 或者两条或两条以上的二次曲线共同构成, 在每两条曲线的拼接处往往会存在角点。所以, 以每条有效轮廓线上的角点为分界点, 分别对有效轮廓线进行分段, 并将各分段轮廓线上像素点分别存放在各自相应的数组内。现以 4-2(b)所示的有效轮廓线为例, 观察该有效轮廓线, 可知其由曲线段(或直线段)拼接而成, 对 4-2(b)所示的两条有效轮廓线进行分段。

在此, 首先需设置如下参数:

(1) 设置一维点结构数组  $tpoint[]$ , 用以先后两次分别存放经过双轮廓线跟踪方法得到的两条有效轮廓线上的所有像素点(由变量  $L$  控制先后跟踪次序);

(2) 设置一维点结构数组  $a1point[]$ ,  $a2point[]$ ,  $a3point[]$ ,  $a4point[]$ , 用以根据判别条件, 分别存放四条分段轮廓线上的像素点;

(3) 设置计数变量 *coun*, *coun1*, *coun2*, *coun3*, *coun4*, *coun5*, *coun6*, *coun7*;

(4) 设置变量 *L*, 初始值赋值为-1, 用以控制跟踪次序, 避免对同一条有效轮廓线进行重复跟踪, 当对一条轮廓线跟踪完毕后,  $L=-L$ ;

(5) 设置方向变量数组 *Direction*[], 并且进行初始化:

```
Direction[8][2]= {{1,0},{1,1},{0,1},{-1,1},{-1,0},{-1,-1},{0,-1},{1,-1}};
```

此外, 因为跟踪过程需要由堆栈辅助八邻域搜索完成, 首先需要定义一个堆栈类, 具体定义如下:

```
class Stack
{
public:
Stack();
~Stack();
bool IsFull();//栈是否已满
void Push(int x, int y);
bool IsEmpty();//栈是否为空
void Pop();//出栈
void GetTop(Point *pt);//取栈顶元素
protected:
private:
Point data[StackSize];//设置结构体数组
int top;
};
```

由堆栈辅助八邻域搜索的代码摘录如下:

```
pStack.Push(StartPoint.Width, StartPoint.Height); //将起始点压入堆栈
//将栈顶元素逐一弹出, 并且依次对其进行八邻域搜索, 直至栈为空
while( !( pStack.IsEmpty() ) )
{
pStack.GetTop(&toppoint);
pStack.Pop();
for(BeginDirect = 0; BeginDirect < 8; BeginDirect++)
{
CurrentPoint.Width=toppoint.Width+Direction[BeginDirect][0];
CurrentPoint.Height=toppoint.Height+Direction[BeginDirect][1];
if((CurrentPoint.Width >= 0 && CurrentPoint.Width < IWidth)
&&(CurrentPoint.Height>=0&&CurrentPoint.Height<IHeight))
{
lpSrc = (unsigned char*)lpDIBBits + ILineBytes *
```

```

        (lHeight-1-CurrentPoint.Height)+CurrentPoint.Width;
Pixel = (unsigned char)*lpSrc;
if(Pixel==0&& !state[CurrentPoint.Height][CurrentPoint.Width])
{
    pStack.Push(CurrentPoint.Width, CurrentPoint.Height);
    state[CurrentPoint.Height][CurrentPoint.Width]=true;
    tpoint[count].Height=CurrentPoint.Height;
    tpoint[count].Width=CurrentPoint.Width;
    count++;
} //endif
} //endif
}
} //endwhile

```

对上面跟踪过程中存放在 *tpoint*[] 中的象素点进行分段存放, 下面按照角点分布的不同, 首先将第一次跟踪到的有效轮廓线进行分段。

如果 *tpoint*[] 中的某象素点满足以下情况中的任意一种, 则将该象素点存放在 *a1point*[] 中:

情况 1:  $tpoint[coun].y \leq jpoint[3].y$ ,  $jpoint[2].x > jpoint[3].x$ ,  $jpoint[4].x > jpoint[5].x$ ,  
 $jpoint[1].y = jpoint[2].y$ ,  $jpoint[3].y = jpoint[4].y$ ,  $jpoint[5].y = jpoint[6].y$ ;

情况 2:  $tpoint[coun].y \leq jpoint[3].y$ ,  $jpoint[2].x < jpoint[3].x$ ,  $jpoint[4].x < jpoint[5].x$ ,  
 $jpoint[5].x > jpoint[6].x$ ;

情况 3:  $tpoint[coun].y \leq jpoint[2].y$ ,  $jpoint[2].x > jpoint[3].x$ ,  $jpoint[4].x > jpoint[5].x$ ;  
 $jpoint[1].y \neq jpoint[2].y$ ,  $jpoint[3].y \neq jpoint[4].y$ ,  $jpoint[5].y \neq jpoint[6].y$ ;

情况 4:  $tpoint[coun].y \leq jpoint[2].y$ ,  $jpoint[2].x > jpoint[3].x$ ,  $jpoint[4].x < jpoint[5].x$ ;

情况 5:  $tpoint[coun].y \leq jpoint[2].y$ ,  $jpoint[2].x > jpoint[4].x$ ,  $jpoint[3].x > jpoint[5].x$ ,  
 $jpoint[5].x < jpoint[6].x$ ;

如果 *tpoint*[] 中的某象素点满足以下情况中的任意一种, 则将该象素点存放在 *a2point*[] 中:

情况 1:  $tpoint[coun].y > jpoint[3].y$ ,  $jpoint[2].x > jpoint[3].x$ ,  $jpoint[4].x > jpoint[5].x$ ,  
 $jpoint[1].y = jpoint[2].y$ ,  $jpoint[3].y = jpoint[4].y$ ,  $jpoint[5].y = jpoint[6].y$ ;

情况 2:  $tpoint[coun].y > jpoint[3].y$ ,  $jpoint[2].x < jpoint[3].x$ ,  $jpoint[4].x < jpoint[5].x$ ,  
 $jpoint[5].x > jpoint[6].x$ ;

情况 3:  $tpoint[coun].y > jpoint[2].y$ ,  $jpoint[2].x > jpoint[3].x$ ,  $jpoint[4].x > jpoint[5].x$   
 $jpoint[1].y \neq jpoint[2].y$ ,  $jpoint[3].y \neq jpoint[4].y$ ,  $jpoint[5].y \neq jpoint[6].y$ ;

情况 4:  $tpoint[coun].y > jpoint[2].y$ ,  $jpoint[2].x > jpoint[3].x$ ,  $jpoint[4].x < jpoint[5].x$ ;

情况 5:  $tpoint[coun].y > jpoint[2].y$ ,  $jpoint[2].x > jpoint[4].x$ ,  $jpoint[3].x > jpoint[5].x$ ,  
 $jpoint[5].x < jpoint[6].x$ ;

经过上述过程，即可将第一条有效轮廓线进行分段，按照相同的方法，将第二条轮廓线进行分段。

如果  $tpoint[]$  中的某像素点满足以下情况中的任意一种，则将该像素点存放在  $a3point[]$  中：

情况 1:  $tpoint[coun].y \leq jpoint[4].y$ ,  $jpoint[2].x > jpoint[3].x$ ,  $jpoint[4].x > jpoint[5].x$ ,

$jpoint[1].y = jpoint[2].y$ ,  $jpoint[3].y = jpoint[4].y$ ,  $jpoint[5].y = jpoint[6].y$ ;

情况 2:  $tpoint[coun].y \leq jpoint[4].y$ ,  $jpoint[2].x < jpoint[3].x$ ,  $jpoint[4].x < jpoint[5].x$ ,

$jpoint[5].x > jpoint[6].x$ ;

情况 3:  $tpoint[coun].y \leq jpoint[5].y$ ,  $jpoint[2].x > jpoint[3].x$ ,  $jpoint[4].x > jpoint[5].x$ ;

$jpoint[1].y \neq jpoint[2].y$ ,  $jpoint[3].y \neq jpoint[4].y$ ,  $jpoint[5].y \neq jpoint[6].y$ ,

情况 4:  $tpoint[coun].y \leq jpoint[4].y$ ,  $jpoint[2].x > jpoint[3].x$ ,  $jpoint[4].x < jpoint[5].x$ ;

情况 5:  $tpoint[coun].y \leq jpoint[5].y$ ,  $jpoint[2].x > jpoint[4].x$ ,  $jpoint[3].x > jpoint[5].x$ ,

$jpoint[5].x < jpoint[6].x$ ;

如果  $tpoint[]$  中的某像素点满足以下情况中的任意一种，则将该像素点存放在  $a4point[]$  中：

情况 1:  $tpoint[coun].y > jpoint[4].y$ ,  $jpoint[2].x > jpoint[3].x$ ,  $jpoint[4].x > jpoint[5].x$ ,

$jpoint[1].y = jpoint[2].y$ ,  $jpoint[3].y = jpoint[4].y$ ,  $jpoint[5].y = jpoint[6].y$ ;

情况 2:  $tpoint[coun].y > jpoint[4].y$ ,  $jpoint[2].x < jpoint[3].x$ ,  $jpoint[4].x < jpoint[5].x$ ,

$jpoint[5].x > jpoint[6].x$ ;

情况 3:  $tpoint[coun].y > jpoint[5].y$ ,  $jpoint[2].x > jpoint[3].x$ ,  $jpoint[4].x > jpoint[5].x$ ;

$jpoint[1].y \neq jpoint[2].y$ ,  $jpoint[3].y \neq jpoint[4].y$ ,  $jpoint[5].y \neq jpoint[6].y$ ,

情况 4:  $tpoint[coun].y > jpoint[4].y$ ,  $jpoint[2].x > jpoint[3].x$ ,  $jpoint[4].x < jpoint[5].x$ ;

情况 5:  $tpoint[coun].y > jpoint[5].y$ ,  $jpoint[2].x > jpoint[4].x$ ,  $jpoint[3].x > jpoint[5].x$ ,

$jpoint[5].x < jpoint[6].x$ ;

### 4.3 曲线类型识别

经过上述处理过程后，已经得到重建旋转曲面的平面曲线  $C$ ，下面将对如何获得旋转体对称轴进行着重介绍。根据本文所研究的三维重建方法，对四条分段曲线进行特征分析，从而根据各分段曲线的类型构造出旋转曲面的对称轴。所以在此之前，必须对四条分段曲线进行类型识别。

现实生活中，由圆弧面结合其他曲面构成的复杂旋转曲面比较多见，如常见的陶瓷工艺制品。本文将着重介绍与分析其母线由弧线与直线或者其他一部分二次曲线拼接而成的旋转体对象的三维重建。上述四个分段数组内存放的像素点分别构成四条曲线段(或直线段)，根据旋转体的对称特性，仅需要对两条有效轮廓线中的一条进行曲线类型识别。下面对构成其中一条有效轮廓线的两条分段曲线段(或直线段)进行如下的类型识别(构成两条分段曲线的像素点分别存放在  $a1point[]$ 、 $a2point[]$  中)：

(1) 设置布尔变量  $line$ ，初始化为  $false$ ；连接分段数组中的起点和终点，构成线段，

该线段的中点如果和存放在该数组中的中间点像素(按照下标次序)重合, 则确认该数组中存放的像素点构成的是直线, 设置布尔变量  $line=true$ , 并且进入  $a2point[]$  对应的分段曲线的类型识别; 否则进入(2)。

部分直线识别代码摘录如下:

```

xb1=(a1point[0].x+a1point[coun4-1].x)/2;
yb1=(a1point[0].y+a1point[coun4-1].y)/2;
if((fabs(a1point[coun4/2].x-xb1)<=0.00001)&&(fabs(a1point[coun4/2].y-yb1)<=0.00001))
{
    line=true;//满足上述判断条件即为直线
    for(count=0; count<coun4; count++)
    {
        lpSrc = (unsigned char *)lpDIBBits + lLineBytes * (lHeight-1-
            a1point[count].y)+a1point[count].x;
        *lpSrc = (unsigned char)0;
    }
} //endif

```

(2)若不满足上述条件, 则假设该曲线段为圆弧, 按照数组下标次序从中选出起点, 中间点和终点分别作为该假设圆弧的起点, 中间点和终点, 根据三点确定一条圆弧的原理, 得到该圆弧的圆心  $(x_r, y_r)$  和半径  $R$ :

$$x_r = \frac{\frac{1}{k_1} \times x_{b2} + y_{b2} + \frac{E_1}{2 \times E_3}}{\frac{1}{k_1} - \frac{E_2}{E_3}} \quad (4.1)$$

$$y_r = -\frac{E_1}{2 \times E_3} - \frac{(x_{b2} + k_1 \times y_{b2} + \frac{k_1 \times E_1}{2 \times E_3}) \times E_2}{E_3 - k_1 \times E_2} \quad (4.2)$$

$$R = \sqrt{(x_1 - x_r)^2 + (y_1 - y_r)^2} \quad (4.3)$$

其中  $(x_1, y_1)$ ,  $(x_2, y_2)$ ,  $(x_3, y_3)$  分别为该假设弧段的起点, 终点, 中间点(按照一维点结构数组  $a1point[]$  中存放的次序计算),  $k_1$  为假设弧段的起点与终点连线的斜率,  $(x_{b2}, y_{b2})$  为假设弧段起点与终点连线的中点坐标,  $E_1 = x_1^2 + y_1^2 - x_3^2 - y_3^2$ ,  $E_2 = x_3 - x_1$ ,  $E_3 = y_3 - y_1$ 。

并且进一步确定该圆弧方程为:

$$(x - x_r)^2 + (y - y_r)^2 = R^2 \quad (4.4)$$

设置布尔变量 *cir*，初始化为 *false*；测试数组中的像素点的坐标值  $(x, y)$  是否均满足方程式 4.4(或者在一定的误差范围内)，若数组中的每个点的坐标值  $(x, y)$  均满足上述条件，则确认该数组中存放的一系列像素点所构成的曲线段为圆弧，并且设置布尔变量 *cir=true*。

部分曲线识别代码摘录如下：

```

rpoint.x=xr;
rpoint.y=yr;
for(counter=0; counter<coun4; counter++)
{
    r=sqrt((rpoint.x-a1point[counter].x)*(rpoint.x-a1point[counter].x)
    +(rpoint.y-a1point[counter].y)*(rpoint.y-a1point[counter].y));
    if(fabs(R-r)<M)
    {
        //M 设置成宏变量
        lpSrc=(unsigned char*)lpDIBBits+lLineBytes*(lHeight-l-a1point[counter].y)
        +a1point[counter].x;
        *lpSrc = (unsigned char)0;
        //对满足上述条件的像素点的计数
        coun8++;
    }
}
//coun4 为对 a1point[]中原有像素点个数的计数
if (coun8==coun4) cir=true;

```

对 *a2point[]* 分别执行上述类型识别，根据旋转体对称性，如果 *a2point[]* 中存放的点构成一条圆弧段，则其对应的 *a4point[]* 中存放的点也构成一条圆弧段。

如图 4-5(b)所示即为经过曲线类型识别后的分段曲线，其中分段线 *C11* 为直线，分段曲线 *C12* 为圆弧，*C* 为圆弧段 *C12* 的圆心。

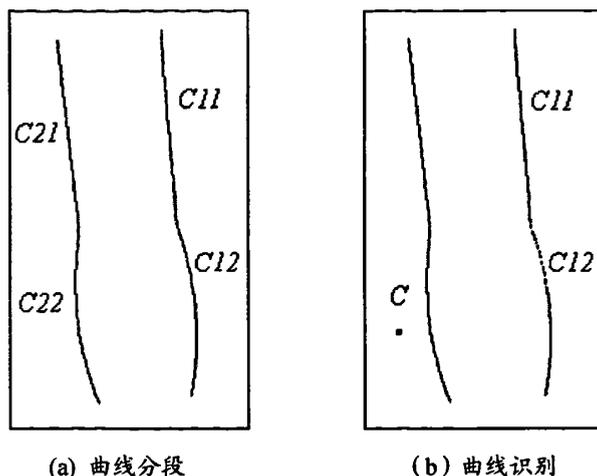


图 4-5 曲线类型识别

Fig.4-5 Curve type detection

#### 4.4 对称轴确定

在确定了构成旋转曲面的母线(也即有效轮廓线)及其分段曲线之后, 还需要确定旋转对称轴。

一般, 相机在对旋转体进行拍摄时, 会出现以下两种情况使得被拍摄的旋转体在影像中并不总是呈现垂直放置的形态: 一是相机本身采用倾斜拍摄; 二是被拍摄的旋转体本身采用倾斜放置。同时, 在图像处理过程中, 往往会由于一些较微弱的误差使得提取出来的两条有效轮廓线不等长。上述一系列情况的出现令对称轴的确定变的尤为复杂。因此, 本文提出了一种基于旋转体对称性原理构造旋转对称轴的方法。

在介绍如何构造旋转对称轴之前, 首先介绍关于特征点的概念。分段数组中的所有像素点构成了相应的分段曲线(也即分段轮廓线), 由于本文主要以母线中含有圆弧段成分的旋转体为研究对象, 根据三点确定一条圆弧的几何特性, 按照下标次序从存放像素点的分段数组中依次取出的第一个元素、中间元素、最后一个元素, 它们分别对应于分段曲线的起点、中间点、终点。下面将这三个点作为每条分段曲线的特征点加以利用。

##### 4.4.1 简单旋转曲面对称轴构造

如图 4-6(a)所示, 为相机拍摄后呈现倾斜状态的旋转体(圆弧面)经过前述一系列处理过程后标记的特征点。如图 4-6(b)所示, 为相机拍摄后呈现理想状态的旋转体经过上述处理过程后标记的特征点。新确定的特征点将通过几何方法共同确定旋转曲面的对称轴。

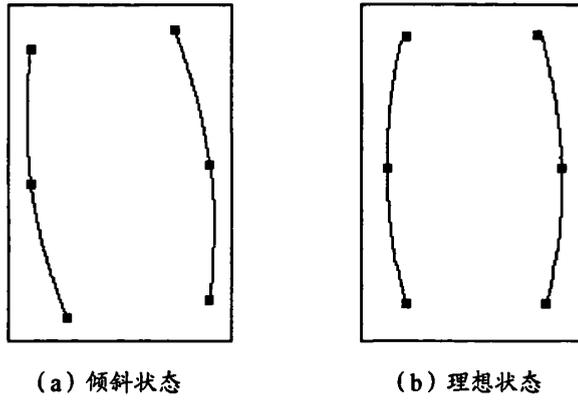


图 4-6 任意形态的旋转体特征点提取

Fig.4-6 Obtain the characteristic points of rotated object in arbitrary status

根据对称原理，右弧可由左弧围绕对称轴旋转一定角度形成，即左弧围绕对称轴旋转一定角度必然与右弧重合，那么左弧上的三个特征点必然与右弧上的三个特征点重合，所以对称轴的确定尤为方便，只需连接左弧与右弧的起点，然后作该连线的中垂线即可。

但是，在图像处理的过程中，往往会由于一些比较微弱的误差使得提取出来的两条弧线不等长，如图 4-7(a)所示，那么由左弧围绕对称轴旋转时其上的三个特征点未必能够与右弧上的三个特征点重合。显然，用以上方法非但不能解决这样的问题，反而会大大降低重建的效果。

为了增强算法的通用性以及重建效果的精确性，本文设计如下方法确定对称轴，如图 4-7(b)所示， $I_1, I_3$  分别为左弧的起点，末点， $I_4, I_6$  分别为右弧的起点，末点。连接  $I_1, I_3$ ，作直线  $I_1I_3$  的中垂线，这里的中垂线交左弧于点  $I_2$ ，交右弧于  $I_5$ 。连接  $I_2$  与  $I_5$ ，并且找出直线  $I_2I_5$  的中点  $I$ ，最后过中点  $I$  作一条平行于直线  $I_1I_3$  的直线  $L$ ，该直线  $L$  即为对称轴。

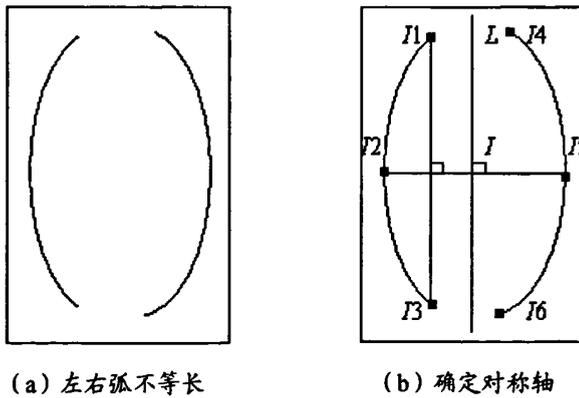
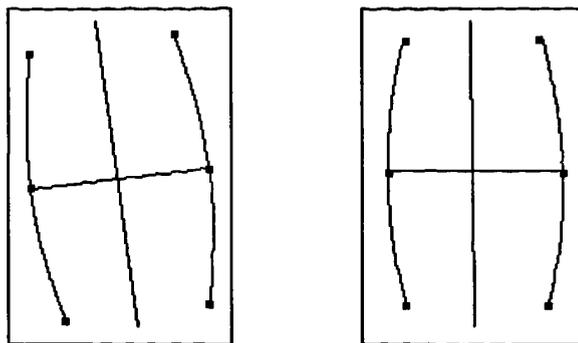


图 4-7 左右弧不等长情况下确定对称轴

Fig.4-7 Construct symmetry axis for two unequal circle arc

对如图 4-6(a)，4-6(b)分别使用上述方法构造对称轴，结果如图 4-8(a)，4-8(b)所示，

其中的标记点为构造对称轴后，新确定的一组对称点。



(a) 倾斜形态的旋转体的对称轴 (b) 理想形态的旋转体对称轴

图 4-8 任意形态旋转体确定对称轴

Fig.4-8 Construct symmetry axis in arbitrary status

#### 4.4.2 复杂旋转曲面对称轴构造

上节中以单一旋转圆弧面为例，对简单旋转曲面的对称轴构造方法进行了介绍。本节将对复杂的旋转曲面的对称轴构造方法进行详细的介绍与分析。现实生活中，许多复杂的旋转曲面都是由简单旋转曲面拼接而成。由于旋转曲面的几何对称性，在构成复杂曲面的多个简单曲面中，只要确定其中之一的对称轴，即可将该对称轴作为整个复杂旋转曲面的对称轴。上节中对旋转圆弧面的对称轴构造进行了介绍，由于三点确定一条圆弧，所以在含有旋转圆弧面的复杂旋转曲面中，利用上节介绍的方法，通过圆弧面去获取整个旋转曲面的对称轴是比较方便的。

复杂的旋转曲面也是由一条平面曲线  $C$  (有效轮廓线) 围绕对称轴旋转而成的。经过上述一系列的图像处理操作后，获得平面曲线  $C$ ，对该曲线进行识别，鉴别出该曲线中是否含有圆弧段成分，具体方法如 4.3 节所述。以 4.3 节中的  $a1point[]$ 、 $a2point[]$  为例，会出现如下两种情况：

(1) 当  $a1point[]$  中存放的像素点构成的是直线时， $a2point[]$  对应的是除圆弧段之外的直线或者其他二次曲线，则可以根据  $a1point[]$  对应的直线确定整个旋转体的对称轴，而这样的对称轴构造相对较易，这里便不予讨论；

(2) 当  $a1point[]$  中存放的像素点构成的是直线或其他二次曲线，同时  $a2point[]$  中存放的像素点构成的是圆弧段时，则可以根据  $a2point[]$  对应的圆弧段确定整个旋转体的对称轴。

总之，只要构成有效轮廓线的两条分段线中含有圆弧段成分，均可以通过圆弧段构造整个旋转体的对称轴。

由于工业、生活中常见的旋转体对象常含有圆弧面，为了增强算法的通用性，如在曲线类型识别时，鉴别的结果为以下两种情况之一，则选择通过圆弧确定整个旋转体的对称轴：

(1)  $line \neq true$ ,  $cir = true$ ;

(2)  $line=true$ ,  $cir=true$ 。

下面即以图 4-9 所示的两条有效轮廓线为例, 对具体的方法以及构造过程中的细节问题进行介绍与分析。

(1) 设置系数变量  $a$ ,  $b$ ,  $c$  和  $d$ ,  $e$ ,  $f$  分别表示两条分段圆弧(按照先后跟踪到的弧段)各自对应的圆弧方程的系数, 计算两条圆弧段的方程系数。

假设弧段 1 的曲线方程为:

$$x^2 + y^2 + a \times x + b \times y + c = 0 \quad (4.5)$$

$(x_1, y_1)$ ,  $(x_2, y_2)$ ,  $(x_3, y_3)$  分别为弧段 1 的起点、终点、中间点;

$$temp\_ex = (x_1 \times y_2 - x_1 \times y_3 - x_2 \times y_1 + x_2 \times y_3 + x_3 \times y_1 - x_3 \times y_2) \quad (4.6)$$

$$a = \frac{(y_3 - y_2) \times (x_1^2 + y_1^2)}{temp\_ex} - \frac{(y_3 - y_1) \times (x_2^2 + y_2^2)}{temp\_ex} - \frac{(y_1 - y_2) \times (x_3^2 + y_3^2)}{temp\_ex} \quad (4.7)$$

$$b = \frac{(x_2 - x_3) \times (x_1^2 + y_1^2)}{temp\_ex} + \frac{(x_3 - x_1) \times (x_2^2 + y_2^2)}{temp\_ex} + \frac{(x_1 - x_2) \times (x_3^2 + y_3^2)}{temp\_ex} \quad (4.8)$$

$$c = -(x_1^2 + y_1^2) - a \times x_1 - b \times y_1 \quad (4.9)$$

假设弧段 2 的曲线方程为:

$$x^2 + y^2 + d \times x + e \times y + f = 0 \quad (4.10)$$

$(x_4, y_4)$ ,  $(x_5, y_5)$ ,  $(x_6, y_6)$  分别为弧段 2 的起点、终点、中间点;

$$temp\_ex = (x_4 \times y_5 - x_4 \times y_6 - x_5 \times y_4 + x_5 \times y_6 + x_6 \times y_4 - x_6 \times y_5) \quad (4.11)$$

$$d = \frac{(y_6 - y_5) \times (x_4^2 + y_4^2)}{temp\_ex} - \frac{(y_6 - y_4) \times (x_5^2 + y_5^2)}{temp\_ex} - \frac{(y_4 - y_5) \times (x_6^2 + y_6^2)}{temp\_ex} \quad (4.12)$$

$$e = \frac{(x_5 - x_6) \times (x_4^2 + y_4^2)}{temp\_ex} + \frac{(x_6 - x_4) \times (x_5^2 + y_5^2)}{temp\_ex} + \frac{(x_4 - x_5) \times (x_6^2 + y_6^2)}{temp\_ex} \quad (4.13)$$

$$f = -(x_4^2 + y_4^2) - d \times x_4 - e \times y_4 \quad (4.14)$$

(2) 确定中垂线方程, 该中垂线垂直且平分弧段 1 的起点到终点的连线  $\overline{I_1 I_3}$ , 其方程为:  $y = k_1 \times (x - m_1) + n_1$ , 其中  $k_1$  为弧段 1 起点和终点连线的垂线的斜率,  $(m_1, n_1)$  为弧段 1 起点和终点连线的中点;

(3) 确定上述中垂线与弧段 1 的交点  $I_2(x_{33}, y_{33})$ , 和弧段 2 的交点  $P_2(x_{44}, y_{44})$ , 以及  $I_2$ 、 $P_2$  的中点  $O(x_{55}, y_{55})$ , 如图 4-9(b)所示;

$$s_1 = 1 / (1 + k_1 \times k_1) \times 0.5 \quad (\text{强制转换成单精度型}) \quad (4.15)$$

$$s2 = -b \times k1 - 2 \times k1 \times n1 + 2 \times k1^2 \times m1 - a + \sqrt{M1 + M2} \quad (4.16)$$

其中:

$$M1 = b^2 \times k1^2 + 2 \times b \times k1 \times a + 4 \times k1 \times n1 \times a - 4 \times k1^2 \times m1 \times a + a^2 - 4 \times k1^2 \times m1^2 \quad (4.17)$$

$$M2 = 8 \times k1 \times m1 \times n1 - 4 \times n1^2 - 4 \times c + 4 \times b \times k1 \times m1 - 4 \times b \times n1 - 4 \times k1^2 \times c \quad (4.18)$$

$$s3 = -e \times k1 - 2 \times k1 \times n1 + 2 \times k1^2 \times m1 - d - \sqrt{M3 + M4} \quad (4.19)$$

其中:

$$M3 = e^2 \times k1^2 + 2 \times e \times k1 \times d + 4 \times k1 \times n1 \times d - 4 \times k1^2 \times m1 \times d + d^2 - 4 \times k1^2 \times m1^2 \quad (4.20)$$

$$M4 = 8 \times k1 \times m1 \times n1 - 4 \times n1^2 - 4 \times f + 4 \times e \times k1 \times m1 - 4 \times e \times n1 - 4 \times k1^2 \times f \quad (4.21)$$

$$x33 = s1 \times s2 \quad (4.22)$$

$$y33 = k1 \times (s1 \times s2 - m1) + n1 \quad (4.23)$$

$$x44 = s1 \times s3 \quad (4.24)$$

$$y44 = k1 \times (s1 \times s3 - m1) + n1 \quad (4.25)$$

$$x55 = (x33 + x44) / 2 \quad (4.26)$$

$$y55 = (y33 + y44) / 2 \quad (4.27)$$

(4) 将弧段 1 的起点、终点的坐标分别另记为 $(x11, y11)$ 、 $(x22, y22)$ , 则过  $O(x55, y55)$  点, 且平行于线段  $\overline{I1I3}$  (弧段 1 的起点和终点的直线) 的直线方程为:  $y = (y22 - y11) / (x22 - x11) (x - x55) + y55$ , 该方程即为对称轴方程;

(5) 如图 4-9(a)所示即是对一个含有圆弧段的旋转曲面的有效轮廓线进行局部特征点标记。对该旋转曲面构造对称轴, 结果如图 4-9(b)所示,  $I1, I3$  分别为弧段 1 上的起点和终点,  $L$  即为旋转体的对称轴,  $I1, I3$  关于对称轴  $L$  的对称点分别为  $P1(x66, y66)$ ,  $P3(x77, y77)$ ;

$$s4 = -e \times k1 - 2 \times k1 \times y11 + 2 \times k1^2 \times x11 - d - \sqrt{M5 + M6} \quad (4.28)$$

其中:

$$M5 = e^2 \times k1^2 + 2 \times e \times k1 \times d + 4 \times k1 \times y11 \times d - 4 \times k1^2 \times x11 \times d + d^2 - 4 \times k1^2 \times x11^2 \quad (4.29)$$

$$M6 = 8 \times k1 \times x11 \times y11 - 4 \times y11^2 - 4 \times f + 4 \times e \times k1 \times x11 - 4 \times e \times y11 - 4 \times k1^2 \times f \quad (4.30)$$

$$s5 = -e \times k1 - 2 \times k1 \times y22 + 2 \times k1^2 \times x22 - d - \sqrt{M7 + M8} \quad (4.31)$$

其中:

$$M7 = e^2 \times k1^2 + 2 \times e \times k1 \times d + 4 \times k1 \times y22 \times d - 4 \times k1^2 \times x22 \times d + d^2 - 4 \times k1^2 \times x22^2 \quad (4.32)$$

$$M8 = 8 \times k1 \times x22 \times y22 - 4 \times y22^2 - 4 \times f + 4 \times e \times k1 \times x22 - 4 \times e \times y22 - 4 \times k1^2 \times f \quad (4.33)$$

$$x66 = s1 \times s4 \quad (4.34)$$

$$y66 = k1 \times (s1 \times s4 - x11) + y11 \quad (4.35)$$

$$x77 = s1 \times s5 \quad (4.36)$$

$$y77 = k1 \times (s1 \times s5 - x22) + y22 \quad (4.37)$$

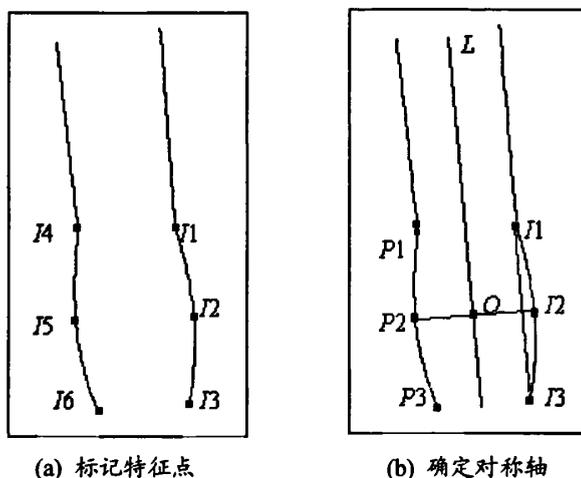


图 4-9 旋转杯体确定对称轴

Fig.4-9 Construct symmetry axis of rotated object

总而言之,对于任意复杂旋转曲面,分析其有效轮廓线,若其有效轮廓线中包含圆弧段成分,那么都可以根据圆弧段,依照上述方法获得整个旋转体的对称轴,这是由旋转体的几何特性决定的。

如图 4-10(a)所示为一只表面为复杂旋转曲面的瓶体经过阈值分割、轮廓跟踪后得到的整体轮廓线;对其进行角点检测及域截取操作后得到如图 4-10(b)所示的两条有效轮廓线;对两条有效轮廓线进行曲线类型识别,鉴别出其中含有圆弧段,分别选取两条圆弧段的起点、中间点、终点做为各自的特征点并加以标记,如图 4-10(c)所示,其中  $C$  为弧段 1 的圆心,  $J1, J2, J3$  为弧段 1 上的三个特征点,  $J4, J5, J6$  为弧段 2 上的三个特征点;根据前述方法,通过构造圆弧段对称轴来获取整个复杂旋转曲面的对称轴,构造结果如图 4-10(d)所示,其中  $L$  即为旋转体的对称轴,  $P1, P2, P3$  分别为弧段 1 上的三个特征点  $J1, J2, J3$  关于对称轴  $L$  的对称点,  $b1$  为  $J1$  及其对称点  $P1$  连线的中点,  $b2$

为  $B$  及其对称点  $P3$  连线的中点，根据对称性， $b1$  点与  $b2$  点必定在对称轴上，并与  $O$  点三点共线。

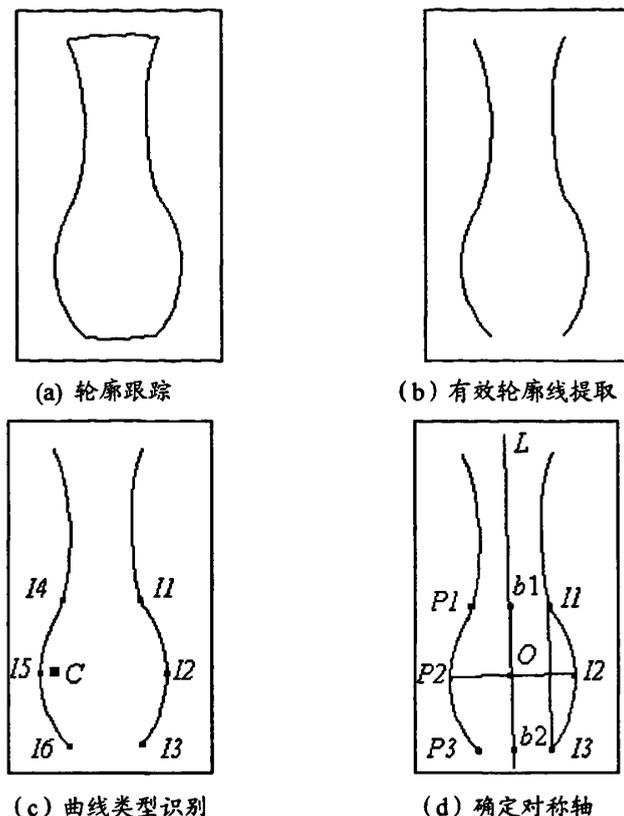


图 4-10 旋转体确定对称轴

Fig.4-10 Construct symmetry axis of rotated object

### 4.5 影像纠正

如图 4-11(a)所示为相机拍摄的旋转杯体经过前述处理方法后得到的两条有效轮廓线，图 4-11(b)所示  $L$  为该旋转体的对称轴， $O$  为左右弧交点的中点，标记点为新确定的一组对称点。为了处理的方便，将图 4-11(b)中的对称轴  $L$  旋转一定的角度，使得  $L$  平行于  $Y$  轴，则对应的弧线及其上所有的像素点坐标将随着对称轴的变化而变化。下面以标记点为例，具体坐标转换过程如图 4-11(c)所示：将对称轴  $L$  围绕中心点  $O$  旋转一定角度使得  $L$  平行于  $Y$  轴，并记新对称轴为  $L1$ ， $I1$  为未旋转之前左弧上的第一个标记点， $d1$  为  $I1$  到  $L$  的距离， $d2$  为  $I1$  到中心点  $O$  的距离，则根据几何特性，经过旋转之后  $I1$  对应的新坐标点  $i1$  到新对称轴  $L1$  的距离应该仍然为  $d1$ ，新坐标点  $i1$  到中心点  $O$  的距离仍然为  $d2$ 。设  $L1: x=x0$ ，中心点  $o(x0,y0)$ ， $i1(x,y)$ ，得方程如下：

$$\begin{cases} x0-x = d1 \\ \sqrt{(x0-x)^2 + (y0-y)^2} = d2 \end{cases} \quad (4.38)$$

解上述方程组得：

$t1(x0-d1, y0-\sqrt{d2^2-d1^2})$ ，根据对称性可得  $t1$  关于  $L1$  在右弧上的对称点  $t4(x0+d1, y0-\sqrt{d2^2-d1^2})$ 。同理可得  $t3(x0-d1, y0+\sqrt{d2^2-d1^2})$ ， $t6(x0+d1, y0+\sqrt{d2^2-d1^2})$ 。设  $I2$  到  $O$  点的距离为  $d3$ ，则易得  $t2(x0-d3, y0)$ ， $t5(x0+d3, y0)$ 。经过上述变换之后，左弧和右弧的位置如图 4-11(d)所示。同理，将存放在一维点结构体数组中的每个象素点进行如上所示的相同原理的坐标转换，并且用转换后的新坐标替换原来的坐标。

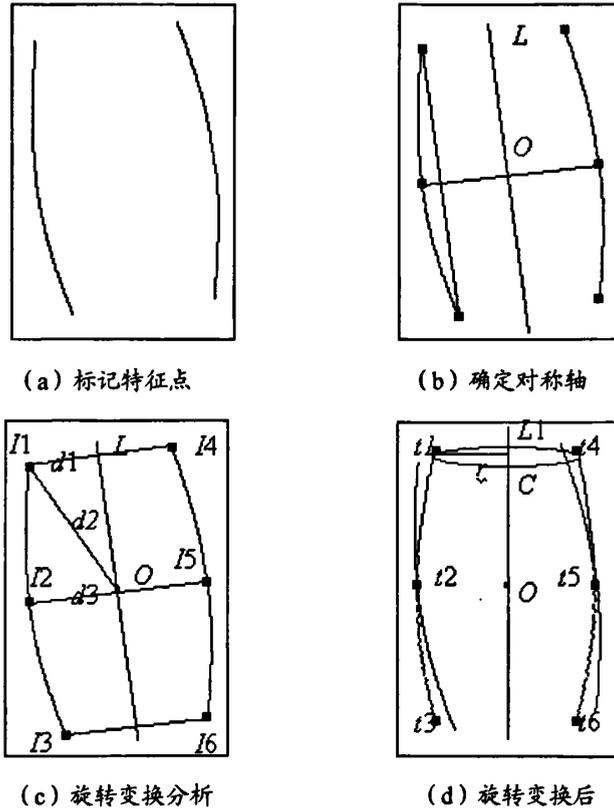


图 4-11 简单旋转曲面的坐标转换

Fig.4-11 Coordinate translation of simple rotated surface

对如图 4-9(b)所示旋转曲面进行坐标转换，原理同上，变换分析如图 4-12(a)所示，转换后结果如图 4-12(b)所示，其中  $t1, t2, t3$  分别为  $I1, I2, I3$  经过变换后的新坐标点， $t4, t5, t6$  为  $p1, p2, p3$  经过变换后的新坐标点。同时，如图 4-12(a)的坐标变换分析对于旋转体上端的直线段部分也同样有效，转换后的效果如图 4-12(b)所示。

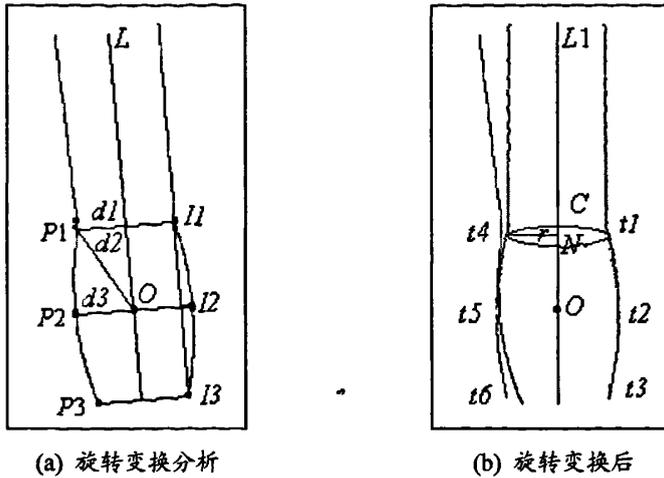


图 4-12 复杂旋转曲面的坐标转换

Fig.4-12 Coordinate translation of complex rotated surface

### 4.6 旋转曲面三维信息获取

图 4-12(b)所示的左边一条有效轮廓线的弧段(以下简称左弧)是由存放在对应点结构体数组中的一系列象素点构成的,当左弧围绕对称轴  $L1$  旋转一周的时候,构成该弧线的每个象素点也围绕对称轴旋转一周,并且形成如图 4-12(b)中空间闭合曲线  $C$  所示的轨迹,该轨迹为一闭合圆弧。所以,由左弧围绕对称轴旋转一周所形成的圆弧曲面即等价于构成左弧的若干个象素点围绕对称轴旋转一周所形成的一系列纬圆的集合。由于对称轴  $L1$  是平行于  $Y$  轴的,假设左弧起点坐标  $t4(x, y)$ ,  $L1: x=x0$ 。过  $t4$  作对称轴  $L1$  的垂线,垂足为  $N(a, b)$ ,  $r$  为纬圆  $C$  的半径,  $N(a, b)$  为纬圆  $C$  的圆心,其中  $r=x0-x$ ,  $a=x0$ ,  $b=y$ 。那么该问题转化为寻找所有满足方程:

$(x-a)^2 + (y-b)^2 + (z-0)^2 = r^2$  ( $y=b$ ) 的点的轨迹,该轨迹是一个闭合的空间纬圆,将上述方程进一步转化为参数方程<sup>[40]</sup>:

$$\begin{cases} x = r \times \sin \varphi \times \cos \theta + a \\ y = r \times \sin \varphi \times \sin \theta + b \\ z = r \times \cos \varphi \end{cases} \quad (4.39)$$

根据以上几何特性,将图 4-12(b)中的闭合曲线  $C$  放置在实际坐标系中。为了理解的方便,暂且将纬圆的圆心  $N$  作为该坐标系的原点,如图 4-13 所示,设纬圆上任意一点  $M$ ,  $\varphi$  是  $\overline{NM}$  与  $Z$  轴的夹角,  $M'$  是  $M$  在  $XOY$  面上的投影,  $\theta$  是  $NM'$  与  $X$  轴的夹角,因为  $M'$  落在  $X$  轴上,所以夹角  $\theta$  为 0, 4.39 方程可进一步转化为如下参数方程:

$$\begin{cases} x = r \times \sin \varphi + a \\ y = b \\ z = r \times \cos \varphi \end{cases} \quad (4.40)$$

其中  $0 \leq \varphi \leq 2\pi$ 。

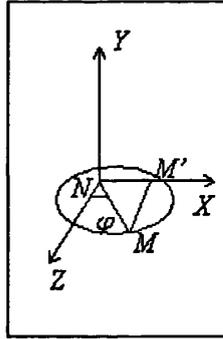


图 4-13 复杂旋转曲面的坐标分析

Fig.4-13 Coordinate analysis of complex rotated surface

上述左弧是有效轮廓线的一部分，其纬圆的计算方法对于有效轮廓线上的其他曲线段(或直线段)部分同样有效。

设置一个二维的点结构体数组  $surfacepoint[ ][ ]$ ，用于存放所有纬圆上的点，每个点对应存放在点结构体数组  $surfacepoint[count][i]$ ，其中下标  $count$  对应于其中一条有效轮廓线上第  $count$  个像素点， $i$  对应于由该点围绕对称轴旋转时得到的第  $i$  个轨迹点。因为需要旋转一周，所以将  $i$  的下限设置为 0，上限设置为 360，即以一度为一个单位进行旋转。这里将其转化为弧度，并且记  $surfacepoint[count][i]$  的三个分量为  $\{nx, ny, nz\}$ ，所以得到旋转曲面上若干个像素点坐标的显式表达为：

$$\begin{cases} nx = a + r \times \sin\left(\frac{3.1416}{180.0} \times i\right) \\ ny = b \\ nz = r \times \cos\left(\frac{3.1416}{180.0} \times i\right) \end{cases} \quad (4.41)$$

#### 4.7 小结

本章提出了基于单幅图像，利用轮廓线进行旋转曲面重建的方法。对从单幅图象中提取出来的旋转体进行分析，根据角点检测序列，辨别提取出有效轮廓线，将有效轮廓线进行分段，并且设置几何条件鉴别各分段曲线的类型；根据各分段曲线的类型及其特征点信息，构造出旋转体的对称轴；最后利用旋转曲面对称特性结合上述特征信息获取旋转曲面的三维信息。这种方法充分利用旋转体的几何特性，既不会出现较大的测量误差问题，也不会出现由于参数精度的高低而影响最终重建效果的情况，实验表明，效果是很好的，可以在文物保护、器皿制作、工业测量等领域有广泛的实用价值。

## 第五章 基于轮廓线的旋转曲面重建系统设计与实现

在第四章中,提出了一种基于轮廓线的旋转曲面三维重建方法。本章将该理论研究成果运用于实践,以 OpenGL 和 VC++6.0 为开发工具,实现一个基于轮廓线的旋转曲面三维重建系统。

### 5.1 使用技术及开发平台

#### 5.1.1 OpenGL技术

(1) 目前图形开发包 DirectX 适于游戏开发以及加强多媒体性能等方面,但只限于 Windows 平台上的开发,而 OpenGL 可以用于跨平台的开发。

(2) 由于 Microsoft 公司在 Win95 以后推出的 Windows 操作系统中提供 OpenGL 图形标准,尤其是三维图形加速卡和微机图形工作站的推出,人们可以在微机上实现三维图形应用,如 CAD 设计、仿真模拟、三维游戏等,从而更有机会、更方便地使用 OpenGL 来建立自己的三维图形世界。

(3) OpenGL 可以与 VC++紧密接合,便于实现有关计算和图形算法,可保证算法的正确性和可靠性。

#### 5.1.2 开发平台

##### (1) 软件平台

本系统在 Windows XP 平台上使用 Visual C++6.0 语言、借助 OpenGL 开发实现。Visual C++6.0 是一个可视化的、集成的开发工具,其采用消息响应机制,具备强大的窗口管理机制和菜单功能,具备很好的人机交互功能。

虽然 OpenGL 在实现图形方面功能强大,但是 OpenGL 在人机交互方面比较薄弱,在本系统中这一点恰恰被 Visual C++6.0 所弥补,所以本系统取得了比较好的实验结果。

##### (2) 硬件平台

Inter(R) Pentium(R)4 CPU 1600MHz(1.60GHz), 256M 的内存, NVIDIA Vanta/Vanta LT (Microsoft Corporation) 显卡(支持 OpenGL)。

### 5.2 系统总体设计

#### 5.2.1 系统设计实现的目标

本系统的设计目标是充分利用旋转曲面的特殊几何性质与相关的图形学理论知识,设计及实现一个基于轮廓线的旋转曲面三维重建系统。具体目标为:

(1) 提供给用户对一个或多个目标区域分布的单幅图像进行轮廓跟踪的方法,并结合经过改进后的自适应 Harris 角点检测方法,方便用户获取重建过程中所必需的有效轮

廓线:

(2) 提供给用户正确构造旋转体对称轴的方法。由于相机拍摄角度或者物体摆放位置的原因, 往往会导致拍摄出来的物体呈现非理想的摆放状态(非垂直摆放); 同时, 由于图像处理时一些较微弱的误差, 使得提取出来的两条有效轮廓线不等长。这些情况的出现会使得构造旋转曲面对称轴变得更加复杂。针对这种情况, 设计出一种确定对称轴的方法, 能够有效地构造出旋转体的对称轴;

(3) 设计相应的坐标转换, 将旋转体纠正为垂直摆放的状态, 以便之后获取旋转体对象的三维信息;

(4) 充分利用旋转曲面的几何特性, 快速提供给用户处理对象的三维几何信息;

(5) 提供光照模型, 来模拟现实世界的光照效果;

(6) 提供友好的操作界面, 方便用户操作;

### 5.2.2 系统的组织结构

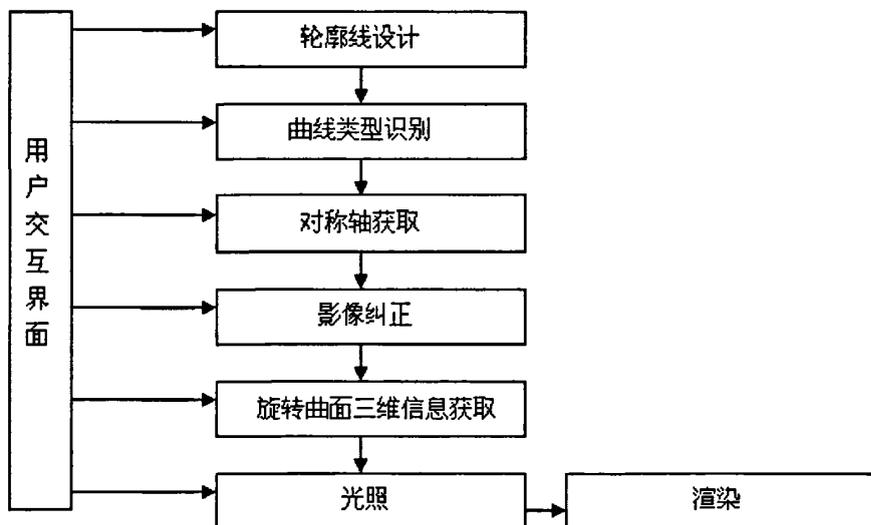


图 5-1 组织结构

Fig 5-1 Organizational structure

### 5.2.3 系统界面规划

系统的界面设计如下图, 包括菜单栏、工具栏、三维视窗口(用于显示三维模型)、二维视窗口(用于有效轮廓线的提取、对称轴的确定等图像处理操作)及状态栏。

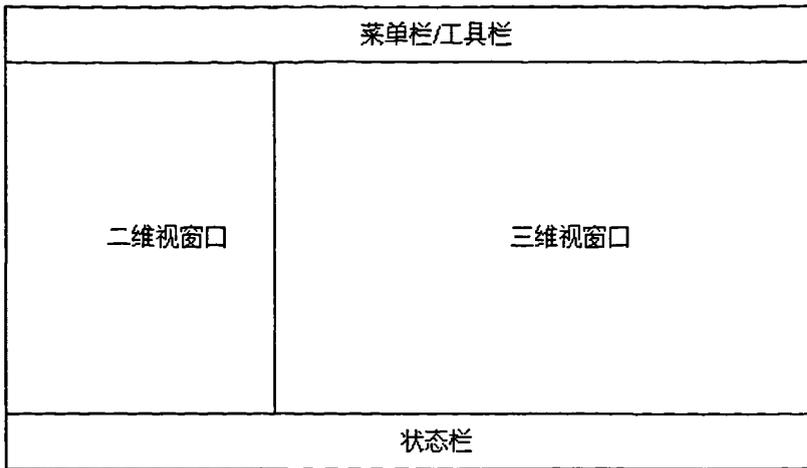


图 5-2 系统整体界面框架

Fig.5-2 The overall interface frame of system

## 5.3 光照渲染

### 5.3.1 光照设置

光照模型用来给三维场景制造光照效果，来增强真实感。通过前面第三章对光照模型的介绍，在本系统中采用 phong 光照模型，加入环境光、漫反射光，镜面光。

光照用来表示材质与光源的相互作用，在设置光源的同时，还需要对材质进行设置，光源位置及材质设置代码摘录如下：

```

GLfloat light_position[] = { 800.0, -800.0, 80.0, 1.0};
glLightfv(GL_LIGHT0, GL_POSITION, light_position);
使用 glEnable(LIGHTING)和 glEnable(GL_LIGHT0)打开光照及启用光源。
GLfloat mat_diffuse[] = { 0.8, 0.0, 0.8, 1.0};
GLfloat mat_ambient[] = { 0.8, 0.8, 0.8, 1.0};
GLfloat mat_specular[] = { 1.0, 1.0, 1.0, 1.0};
GLfloat mat_shininess[] = { 15.0};//镜面指数
glMaterialfv(GL_FRONT, GL_AMBIENT, mat_ambient);
glMaterialfv(GL_FRONT, GL_SPECULAR, mat_specular);
glMaterialfv(GL_FRONT, GL_SHININESS, mat_shininess);

```

### 5.3.2 曲面网格划分及法向量

在材质接收光照之前，需要正确计算顶点的法向量。曲面上各顶点的法向量计算方法有许多，若各顶点是根据方程建立起来的，则可以根据函数表达式求偏导的方法获得其法向量。

本系统是使用 OpenGL 图形加速库开发而成，在利用 OpenGL 进行旋转体表面的重

建时, 需要首先将旋转表面上的点进行网格化。按照右手规则, 将  $surfacepoint[][]$  中的点集划分为由若干平面小四边形组成的集合, 划分格式如下: 由  $(count, i)$ ,  $(count+1, i)$ ,  $(count+1, i+1)$ ,  $(count, i+1)$  构筑一个平面小四边形, 这样每个点可以参与构筑平面四边形, 最终完成网格化。

法向量是曲面上某点处垂直于该面的向量, 不但是得到理想光照效果的重要条件, 也是产生逼真图形的关键因素, 它通过确定物体相对光源的方位, 可计算出此点的受光情况。对于一个平面, 其上每点法向量都一样。而对于曲面, 事实上每点处的法向量都不一样。在本文中, 由于旋转表面上的点满足方程:  $(x-a)^2+(y-b)^2+(z-0)^2=r^2$ , 故可以根据偏导公式求得每个象素点的法向量。以二维点结构体数组元素  $surfacepoint[count][i]$  为例, 法向量  $FX[count][i]$  的三个分量记为  $\{sx, sy, sz\}$ , 该元素所对应的象素点法向量即为:

$$\begin{cases} sx = 2 \times nx \\ sy = 2 \times ny \\ sz = 2 \times nz \end{cases} \quad (5.1)$$

其中,  $\{nx, ny, nz\}$  是  $surfacepoint[count][i]$  的三个分量。设置二维点结构体数组  $FX[][]$  用于存放平面小四边形上四个顶点的法向量。依次计算出  $surfacepoint[count][i]$  和  $surfacepoint[count+1][i+1]$  的法向量, 将这两个对角线顶点的法向量相加并求平均法向量, 然后将该平均法向量分别赋值给平面小四边形的四个顶点。

编程时, 在每次绘制小四边形面片之前, 对要绘制的四个点赋值法向量, 这样就可以在绘制四边形表面的同时将法向量信息计算完, 再通过  $glNormal3f(R1, R2, R3)$  指令完成法向量的显示。最后在正射投影的假设下, 引进光照, 观察实验效果。

对第四章中图 4-10 所研究的旋转瓶体进行三维重建, 为了增强重建结果的真实感, 编程实现时, 将杯体旋转一定角度, 结果如图 5-3 所示。

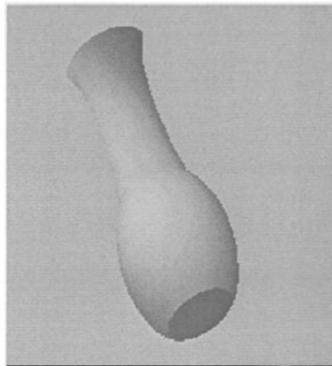


图 5-3 旋转体三维重建

Fig.5-3 3D reconstruction of rotated object

## 5.4 交互功能

系统提供给用户对鼠标的简易交互操作。为了增强瓶体的真实感效果,在三维视图中设置物体坐标系,该坐标系是相对自身移动的。将物体坐标系统绕三坐标旋转的角度和坐标分别存放在 *sceneRot*[3], *scenePos*[3]中,用户按下鼠标左键拖动即可对三维模型进行旋转。

## 5.5 实验

### 5.5.1 实验结果

为了验证基于轮廓线的旋转曲面重建系统的有效性 with 正确性,利用实际物体进行了以下三组实验。

如图 5-4 所示,分隔栏左上角为一只由圆弧面构成的茶杯,由 Sony707 数码照相机拍摄得到,因相机是手工拍摄,所以拍摄出来的茶杯呈现倾斜放置的状态。分隔栏右边则是经过上述系统实现的重建效果。编程实现时,将杯体旋转一定角度,以增强真实感。

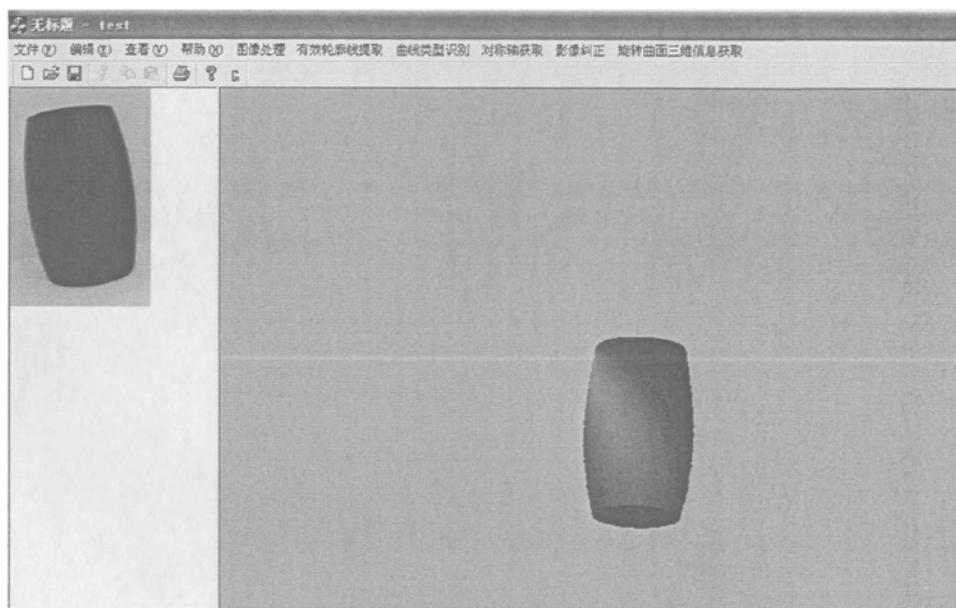
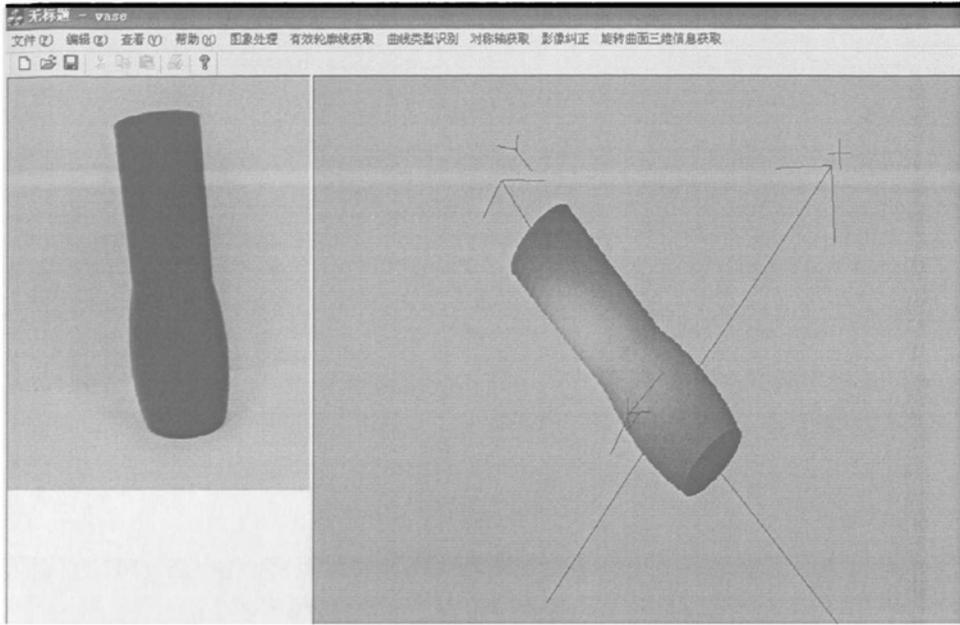


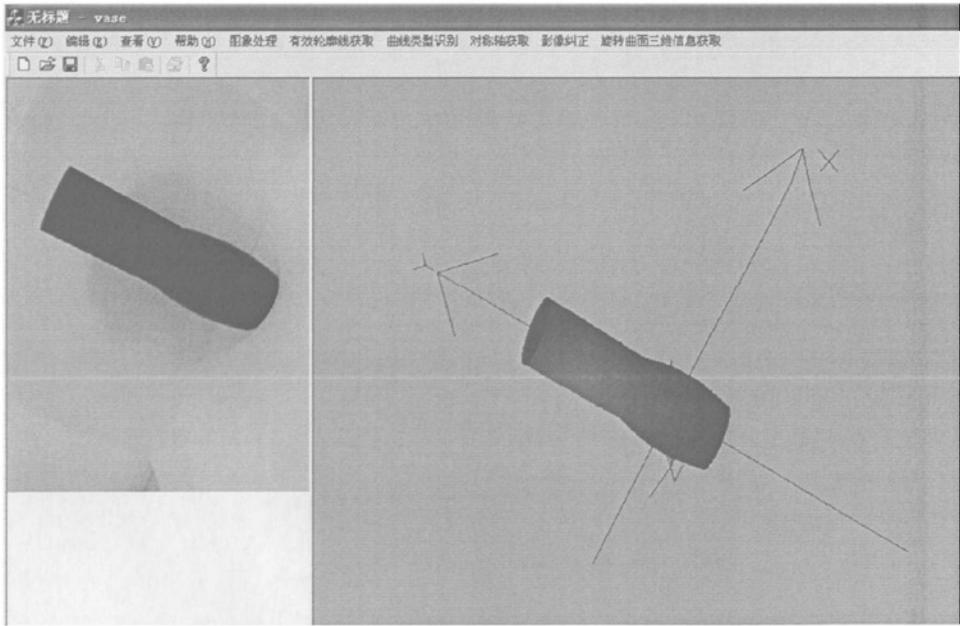
图 5-4 旋转杯体三维重建

Fig.5-4 3D reconstruction of cup

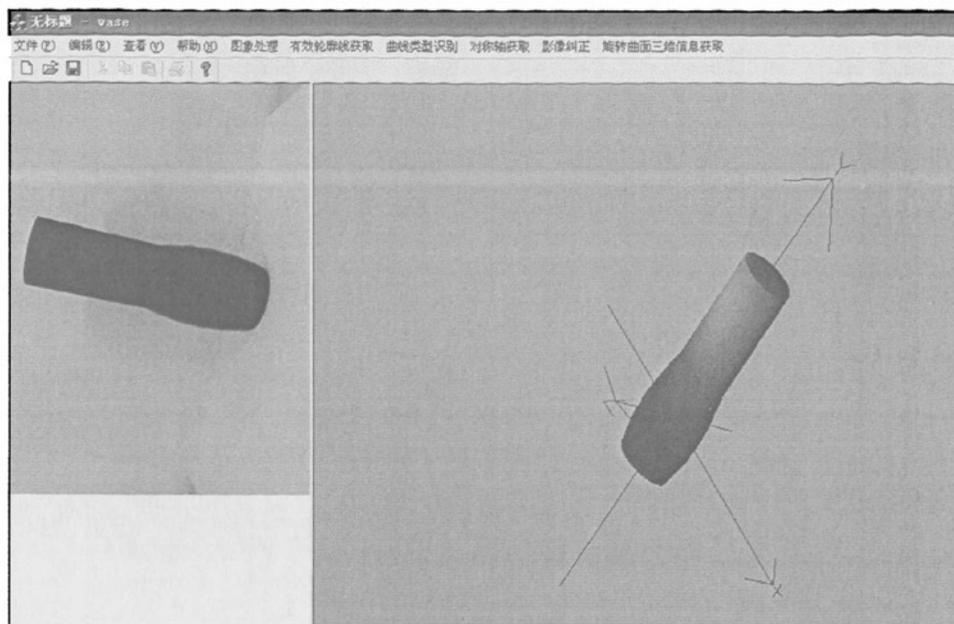
如图 5-5(a)所示,为实现复杂旋转曲面三维重建的系统界面(设置物体坐标系)。分隔栏左上角为一只由柱面和圆弧面拼接而成的花瓶、由 Sony707 数码照相机拍摄得到,分隔栏右边则为经过上述系统实现的重建效果。图 5-5(b)、5-5(c)分别显示了呈现不同摆放状态的花瓶经过该系统实现重建的效果。



(a) 任意倾斜状态



(b) 任意倾斜状态



(c) 任意倾斜状态

图 5-5 任意状态摆放花瓶的三维重建

Fig.5-5 3D reconstruction of vase in arbitrary status

如图 5-6 所示，分隔栏左上角为一只表面由二次曲面和圆弧面拼接而成的花瓶，分隔栏右边则为经过上述系统实现的重建效果。

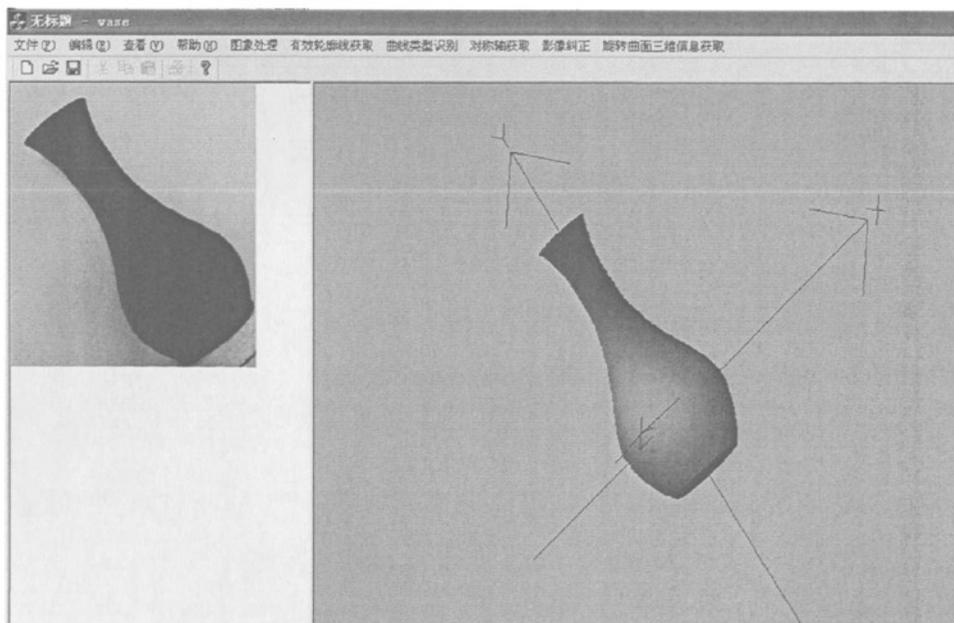


图 5-6 花瓶的三维重建

Fig.5-6 3D reconstruction of vase

### 5.5.2 误差分析

对上述三组实验中的实验 1、实验 2(a)、实验 3 分别进行误差分析，结果如下所示：

	原旋转杯体	重建后旋转杯体	误差
上径	1.60cm	1.60cm	0.0mm
下径	1.60cm	1.60cm	0.0mm
高	3.0cm	2.98cm	0.2mm

(a) 误差分析

	原旋转瓶体	重建后旋转瓶体	误差
上径	1.50cm	1.45cm	0.5mm
下径	1.50cm	1.45cm	0.5mm
高	5.3cm	5.2cm	1.0mm

(b) 误差分析

	原旋转瓶体	重建后旋转瓶体	误差
上径	1.10cm	1.00cm	1.0mm
下径	1.58cm	1.54cm	0.4mm
高	4.8cm	4.78cm	0.2mm

(c) 误差分析

图 5-7 误差分析

Fig.5-7 Errors analysis

## 5.6 小结

根据 5.5 节中三维重建系统生成的实验效果及误差分析可见, 本文所提出的基于轮廓线的旋转曲面三维重建方法可以根据旋转体的二维灰度图像方便有效的重建出其三维形状。

文献[13]提出了一种基于影像和物方重建的旋转体三维重建方法, 并且依照该方法构建了相应的曲面重建系统。通过系统的具体实现方法及其生成的实验效果、误差分析, 将文献[13]所提出的旋转体重建方法与本文所提出的旋转体重建方法进行如下的比较:

	文献方法	本文方法
影像获取	旋转平台(包括平面格网板、能够驱动平板旋转的步进电机), CCD 相机, 由计算机控制旋转平台旋转和 CCD 相机获取黑白影像	Sony707 数码照相机获取影像
有效轮廓线	通过影像上左右对称的成像计算出旋转体的轮廓线(前提: 相机的主光轴指向并垂直于旋转体的对称轴), 涉及参数较多	根据旋转体倾斜角度, 结合角点检测序列, 从整体闭合轮廓线中提取有效轮廓线, 涉及参数较少
旋转对称轴	基于物方解的方法直接计算出旋转体上口的圆心坐标, 对称轴由该圆心坐标唯一确定(前提: 旋转体对称轴必须垂直于平面格网板)	根据有效轮廓线的分段曲线及其特征信息构造旋转体对称轴, 适用于呈现任意摆放状态旋转体(旋转体对称轴无需垂直于参照水平面)
三维重建效果	对旋转体进行重建的结果与游标卡尺测量的结果进行比较, 测量误差在 1~1.5mm(主要比较的是旋转体的底面和顶面直径、旋转体的高)	基于单幅图像的操作, 不受太多外在因素的干扰, 所涉及参数较前者而言较少, 测量误差在 0.2~1.0mm 之间

图 5-8 两种旋转体重建方法比较

Fig.5-8 Comparison between two 3D reconstruction methods of rotated surface

由此可见, 本文所提出的基于轮廓线的旋转曲面重建方法, 根据旋转曲面自身所具备的特殊几何特性, 方便快捷的实现旋转体表面的三维重建。不仅有效避免了一般重建方法所要求的需要较多的几何参数去精确表示曲面形态的缺陷, 而且一定程度上提高了重建的精度。

## 第六章 结论和展望

### 6.1 总结

本文在广泛阅读了国内外关于旋转曲面重建的文献资料基础之上,对由二维灰度图像重建出其三维形状进行了研究,提出了一种基于轮廓线进行旋转曲面重建的方法。经过前面五章的论述,我们提出了一种基于轮廓线的旋转曲面三维重建方法,并在此基础上设计并实现了一个旋转曲面三维重建系统,达到了较为理想的效果。论文的主要研究工作体现在以下几个方面:

#### 1、轮廓线设计方面

(1) 有效轮廓线提取:通过单轮廓线跟踪算法得到整个旋转体的闭合轮廓线,利用角点检测算法,获取角点检测序列,并由此提取出在本课题研究背景下的三维重建过程所需要有效轮廓线;

(2) 有效轮廓线特征信息获取:利用双轮廓线跟踪法,对两条有效轮廓线分别进行轮廓线跟踪。并且以每条轮廓线上的中间一个角点为分界点,按照不同类型对轮廓线进行分段,并将各分段轮廓线上像素点的坐标信息分别存放在各自相应的数组内;

#### 2、对称轴构造方面

(1)根据本文所研究的三维重建方法的要求,对四条分段曲线分别进行特征分析,由于本文主要以母线中含有弧线段成分的旋转体为研究对象,所以根据几何性质设置判断条件来识别分段曲线的类型,从而可以依照分段曲线的类型构造旋转体的对称轴;

(2)根据上述判定结果,结合分段曲线的特征信息,利用本文所提出的构造方法构造旋转曲面的对称轴;

#### 3、影像纠正方面

相机拍摄的旋转体往往呈现非理想状态(非垂直摆放),为了处理的方便,设计相应的坐标变换,将旋转体纠正为垂直摆放的状态,以便之后获取旋转体对象的三维信息。

#### 4、旋转曲面三维信息获取方面

充分利用旋转曲面的几何特性,结合对称轴与有效轮廓线的特征信息,获取旋转体对象的三维信息。

#### 5、真实感图形技术与 OpenGL 技术

光照模型用来给三维场景制造光照效果,来增强真实感。在本文中采用合适的光照模型,加入环境光、漫反射光,镜面光,以实现对场景模型的照明,并结合 OpenGL 技术,实现三维物体的显示。

### 6.2 展望

近年来,国内外针对旋转曲面重建做了很多的研究与实践工作,而在这其中,如何

快速精确的实现由二维灰度图像重建出旋转体的三维模型更是一个具有挑战性的课题。我们的工作还仅限于实验室研究阶段，为了使现有的重建技术走向成熟、达到先进、经济和使用的目的，需要从以下方面进行更深入的研究：本文提出了基于单幅图像，利用轮廓线进行旋转曲面重建的方法。由于圆弧面旋转体常见于日常生活中，为了体现实际应用的价值，这里所涉及的轮廓线多为直线或者是圆弧，而在现实应用中，除了特殊曲线曲面(直线、圆弧、圆柱、球面等)之外，自由曲线曲面(四边域样条曲线)、不规则曲线也可见于工业产品的几何外形中。本系统主要针对一些常见的旋转曲面进行重建，并详细阐述了相关的重建方法。对于一些复杂曲面的构建，譬如自由曲面的重建，随着曲面的复杂化，重建过程所涉及到有效轮廓线提取以及旋转轴确定也将更加复杂化，在今后的研究工作中，将进一步拓展重建对象的领域，加强对旋转曲面重建算法的深入研究。

## 致 谢

这篇论文是我的课题曲面重建——基于轮廓线的旋转曲面三维重建的理论实践研究总结。由于自身的学识水平所限，再加上现代高新技术日新月异的发展速度，我的研究很可能是粗浅的，论文也难免有不足乃至错误之处，恭请老师和专家不吝赐教。

首先，我要感谢我的导师林意副教授。两年里，他为我的学习和成长，给予了精心的指导、倾注了大量心血。他治学严谨，为我提供了许多向同行专家请教学习的机会，要求我走出门去多交流，埋下头来做学问，着力培养思考能力、实践能力和创新能力。他用个人的精神风范影响我的成长进步。在此，我要向林意副教授致以崇高的敬意和深深的感激之情！

其次，要感谢其他授课的老师，他们教授了我丰富的知识，给予了我生活学习无比的关怀和支持，使我两年的研究生生活丰富多彩。

最后，我要感谢一起渡过两年美好时光的同学们。在生活上的互帮互助，在学术上的广泛探讨，使我在学校里愉快的度过了人生重要的阶段。

## 参考文献

1. 寿伟信. 多层螺旋 CT 曲面重建在肋骨不全骨折中的应用[J]. 浙江临床医学, 2007, 9(9):1271.
2. 张剑清, 郑丽. 基于结构光的不规则工业钣金件三维曲面重建[J]. 地理空间信息, 2004, 2(6):9—10.
3. 彭翔, 高鹏东, 刘晓俐, 等. 真实感人脸模型的细分曲面重建[J]. 计算机辅助设计与图形学学报, 2006, 18(5):742—746.
4. 王越. 空间曲面三维重建关键技术的研究[D]: [硕士学位论文]. 哈尔滨: 哈尔滨工业大学, 2004.
5. 蔡钦涛. 基于图像的三维重建技术研究[D]: [硕士学位论文]. 杭州: 浙江大学, 2004.
6. 李介谷. 计算机视觉的理论和实践[M]. 上海: 上海交通大学出版社, 1991. 21—23.
7. 徐奕, 周军, 周源华. 立体视觉匹配技术[J]. 计算机工程与应用, 2003(15):1—3.
8. 章毓晋. 图像理解和计算机视觉[M]. 北京: 清华大学出版社, 2000. 108—140.
9. 张祖勋, 张剑清. 数字摄影测量学[M]. 武汉: 武汉测绘科技大学出版社, 1996. 24—32.
10. 王之卓. 摄影测量原理[M]. 北京: 测绘出版社, 1979. 34—42.
11. Wong K Y K, Mendonca P R S, Cipolla R. Reconstruction of Surfaces of Revolution from Single Uncalibrated Views[C]. British Machine Vision Conference, British, 2002.
12. Utcke S, Zisserman A. Projective Reconstruction of Surfaces of Revolution[C]. DAGM03, Magdeburg, Germany, 2003.
13. 郑顺义, 徐刚, 张祖勋, 张虎龙. 旋转体三维重建方法研究[J]. 武汉大学学报, 2006, 31(12): 1047—1050.
14. M. Bichsel, A. P. Pentland. A simple algorithm for shape from shading. IEEE Proceedings of Computer Vision and Pattern Recognition, 1992.
15. Ruo Zhang, Ping-Sing Tsai, James Cryer, Shape from Shading[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1999, 21(8):690—706.
16. 章毓晋. 图象处理和分析[M]. 北京: 清华大学出版社, 1998. 72—85.
17. 章毓晋. 图象处理和分析[M]. 北京: 清华大学出版社, 1998. 179—197.

18. 求是科技. 数字图像处理典型算法及实现[M]. 北京: 人民邮电出版社, 2006. 406—408.
19. 赵文彬, 张艳宁. 角点检测技术综述[J]. 计算机应用研究, 2006, 10(2): 17—19.
20. Rosenfeld A, Weszka J S. An Improved Method of Angel Detection on Digital Curve[J]. IEEE Trans. Computers, 1975, C-24(9): 940—941.
21. Freeman H, Davjs L S. A Corner Finding Algorithm for Chain Coded Curves[J]. IEEE Trans. Computers, 1977, 26(3): 297—303.
22. Asada H, Brady M. The Curvature Primal Sketch[J]. IEEE Trans. PAMI, 1986, 8(1): 2—14.
23. Mediono G, Yasumoto Y. Corner Detection and Curve Representation Using Cubic B-splines[J]. Computer Vision, Graphics and Image Processing, 1987, 39(3):267—278.
24. Witkin A P. Scale Space Filtering[C]. Karlsruhe: Proceedings of the 8<sup>th</sup> International Joint Conference on Artificial Intelligence, 1983, 1019—1021.
25. Koenderink J J. The Structure of Image[J]. Biological Cybernetics, 1984, 50(5):363—370.
26. Deriche R, Giraudon G. A Computatinal Approach for Corner and Vertex Detection[J]. Computer Vision, 1993, 10(2):95—102.
27. 黄宣达. 摄像机标定中亚像素级角点检测算法[J]. 计算机与数字工程, 2008, 36(4): 137—140.
28. 谢东海, 詹总谦, 江万寿. 改进 Harris 算子用于点特征的精确定位[J]. 测绘信息与工程, 2003, 28(2): 22—23.
29. 张小洪, 李博, 杨丹. 一种新的 Harris 多尺度角点检测[J]. 电子与信息学报, 2007, 29(4): 1735—1738.
30. 张登荣,刘辅兵,俞乐, 蔡志刚, 邓超. 基于 Harris 算子的遥感影像自适应特征提取方法[J]. 国土资源遥感, 2006(2):35—38.
31. 赵万金, 龚声蓉, 刘纯平等. 一种自适应的 Harris 角点检测算法[J]. 计算机工程, 2008, 34(10), 212—217.
32. 刘阳成, 朱枫, 一种新的棋盘格图像角点检测算法[J]. 中国图像图形学报, 2006, 11(5): 656—660.
33. 王玉珠, 杨丹, 张小洪. 基于 B 样条的改进型 Harris 角点检测算法[J]. 计算机应用研究, 2007(2), 192—194.

34. David F. Rogers. 计算机图形学的算法基础[M]. 机械工业出版社, 2000.
35. C.M.Goral, K.E. Torrance, D.P.Greenberg, and etc, Medeling the interation of light between Diffuse Surfaces[J].Computer Graphics, 1984,18(3).
36. 乔林, 费广正等, OpenGL 程序设计[M]. 北京:清华大学出版社, 2000.
37. Dave Shreiner, Mason Woo, Jakie Neider ,and etc. OpenGL 编程指南[M]. 北京: 人民邮电出版社, 2005.
38. 僧德文, 李仲学, 李春民, 李翠平. 基于 OpenGL 的真实感图形绘制技术及应用[J]. 计算机应用研究. 2005, 22(3): 173—175.
39. 李颖, 薛海斌, 朱伯立等. OpenGL 技术应用实例精粹[M]. 北京: 国防工业出版社, 2001, 1—5.
40. 同济大学应用数学系. 高等数学[M]. 北京: 高等教育出版社, 2002. 104—106.

## 攻读硕士学位期间发表的论文

- 1 .钱苏斌. 基于轮廓线的旋转曲面三维重建[J].计算机工程与设计,已录用(2009年10月刊发).