

中图分类号: TP311  
学科分类号: 081203

论文编号: 1028704 09-0109

# 硕士学位论文

## 基于手绘草图的三维模型检索系统

硕士生姓名	张必欢
学科、专业	计算机应用技术
研究方向	智能人机交互
指导教师	谢强 副教授

南京航空航天大学

研究生院 信息科学与技术学院

二〇〇八年十二月

Nanjing University of Aeronautics and Astronautics  
The Graduate School  
College of Information Science and Technology

# 3D Model Retrieval System Based On Freehand Sketch

A Thesis in

Computer Science and Technology Engineering

By

Zhang Bihuan

Advised by

Xie Qiang

Submitted in Partial Fulfillment

of the Requirements

for the Degree of

Master of Engineering

December, 2008

# 承诺书

本人郑重声明：所呈交的学位论文，是本人在导师指导下，独立进行研究工作所取得的成果。尽我所知，除文中已经注明引用的内容外，本学位论文的研究成果不包含任何他人享有著作权的内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明。

本人授权南京航空航天大学可以有权保留送交论文的复印件，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、缩印或其他复制手段保存论文。

(保密的学位论文在解密后适用本承诺书)

作者签名：\_\_\_\_\_

日 期：\_\_\_\_\_

## 摘要

数字多媒体技术、计算机视觉技术、三维图形硬件的飞速发展拓展了三维图形的应用前景。特别是在互联网上已经有大量共享的三维模型。越来越多的人希望能够借助某种工具，在互联网上或者数据库中快速得到自己需要的三维模型。因此，研究和开发基于手绘草图的三维模型检索具有重要的理论意义和实际价值。本文的工作主要包含以下几个方面：

(1) 第一章阐述了三维模型检索的主要研究内容和研究意义，分析了三维模型检索领域的相关技术，并对当前较为典型的三维模型检索系统进行了介绍。

(2) 第二章介绍本文的框架结构。首先介绍基于手绘草图是三维模型检索系统的体系结构，其次介绍关键处理流程，然后介绍本文的关键技术，最后介绍性能评估方法。

(3) 第三章和第四章研究适合三维模型检索的草图识别和草图特征提取方法。经过冗余点消除、聚点消除、笔画拟合等草图预处理方法以后，识别用户输入的草图。草图被分为外围轮廓笔画和内部笔画。草图的外围轮廓包含大量的信息，本文使用四种轮廓描述方法描述草图的轮廓特征。内部笔画被识别为六种基本图元，然后用基本图元的特征描述内部笔画的特征。

(4) 第五章研究三维模型特征向量库生成方法和相似性计算方法。其中，三维模型特征向量库生成方法包括三个步骤：首先，通过三视图正交投影方法获取三维模型的投影轮廓；其次，提取轮廓的特征向量；然后，把特征向量存入系统特征向量库。

(5) 最后基于上文的研究，实现基于手绘草图的三维模型检索系统。

**关键词：**手绘草图，草图识别，傅利叶形状描述，三维模型检索，轮廓特征

## Abstract

Improved digital multimedia technology, computer vision theory and graphics processing unit have expanded the applications of 3D graphics. Especially, a large number of 3D models are available from internet. People often want to use a tool to help them find desirable 3D models from Internet or model databases efficiently and accurately. So researching and developing of the Freehand Sketch-based 3D model retrieval system with good performance has an important theoretical and practical value. The main work can be concluded as following:

(1) In chapter I, The main content of 3D model retrieval and its great importance are introduced, some relevant technology about 3D model retrieval is analyzed and some typical 3D model retrieval system is enumerated.

(2) In chapter II, the framework structure of this paper is introduced. First, the framework of 3D model retrieval system based on freehand sketch is introduced; second, the key data flow chart is introduced; third, the key technologies of this paper are introduced; finally, the method of retrieval performance evaluation standard is introduced.

(3) In chapter III and chapter IV, the methods of sketch recognition and sketch feature extraction are studied, which are suited for 3D model retrieval. After eliminating redundant points, eliminating accumulation points, fitting stroke and so on sketch pre-processing methods, the peripheral contour and the inside strokes can be recognized. The peripheral contour contains large amount of information, so four kinds contour description methods are used to describe the freehand sketch contour characteristic. The inside strokes are identified six kinds element figures, and the inside strokes are described with the element figures.

(4) In chapter V, 3D model feature vector database generation and similarity calculation method are studied. Therein, generating 3D model feature vector database contains three steps: First, the projection contours of 3D model are get by three-plane orthogonal projection method; second, the feature vectors of the projection contours are extracted; third, the feature vectors are inserted into system feature vector database; finally, use similarity calculation method to retrieval user need 3D model.

(5) Finally, Based on the above study, a 3D model retrieval system based on freehand sketch is designed.

**Keywords:** Freehand sketch, sketch recognition, Fourier shape description, 3D model retrieval, contour characteristic

# 目 录

第一章 绪论.....	1
1.1 研究背景.....	1
1.1.1 三维模型建模技术.....	1
1.1.2 手绘草图技术.....	1
1.1.3 三维模型检索现状.....	2
1.1.4 现有的三维模型检索系统简介.....	3
1.2 三维模型检索系统的一般框架.....	4
1.3 研究的内容及其意义.....	5
1.4 本文的工作及其组织结构.....	6
第二章 基于手绘草图的三维模型检索系统总体框架.....	7
2.1 系统的体系结构.....	7
2.2 系统处理流程.....	8
2.2.1 草图识别数据流程.....	8
2.2.2 草图特征提取流程.....	9
2.2.3 三维模型特征提取流程.....	10
2.3 关键技术.....	11
2.3.1 三维模型检索系统中的草图识别.....	11
2.3.2 草图外围轮廓特征提取.....	12
2.3.3 系统特征向量库生成.....	12
2.3.4 相似性计算方法.....	12
2.3.5 相关反馈.....	13
2.3.5.1 相关反馈概念.....	13
2.3.5.2 相关反馈算法.....	14
2.3.5.3 用户反馈信息.....	16
2.4 检索性能评估.....	17

2.5 本章小结.....	17
<b>第三章 三维模型检索系统中的草图识别.....</b>	<b>18</b>
3.1 草图预处理.....	18
3.1.1 冗余点消除 .....	18
3.1.2 聚点消除 .....	19
3.1.3 笔画的拟合 .....	19
3.1.4 笔画闭合处理 .....	20
3.2 草图外围轮廓提取.....	20
3.2.1 外围轮廓提取方法 .....	21
3.2.2 平滑处理 .....	22
3.3 草图基本图元识别.....	23
3.3.1 相关计算 .....	23
3.3.2 多基本图元类型 .....	23
3.3.3 基本图元识别流程 .....	24
3.4 本章小节.....	26
<b>第四章 草图外围轮廓特征提取 .....</b>	<b>27</b>
4.1 基于等点采样的轮廓特征.....	27
4.1.1 等点采样方法描述 .....	27
4.1.2 特征提取匹配方法描述 .....	28
4.1.3 特征匹配 .....	29
4.2 基于极图谱的轮廓特征.....	30
4.2.1 Delaunay 图的关联图谱表示 .....	30
4.2.2 极图谱特征提取 .....	33
4.3 基于离散傅立叶变换的轮廓特征.....	35
4.3.1 傅立叶变换方法描述 .....	35
4.3.2 不变性处理 .....	35
4.3.3 维数选择 .....	36

4.3.4 特征向量的生成 .....	38
4.3.5 实验结果分析 .....	38
4.4 基于多边形的轮廓特征 .....	39
4.4.1 多边形近似 .....	39
4.4.2 多边形特征提取 .....	40
4.5 本章小节 .....	41
<b>第五章 三维模型轮廓特征提取及相似性计算 .....</b>	<b>42</b>
5.1 三维模型预处理技术 .....	42
5.1.1 预处理的必要性 .....	42
5.1.2 坐标系原点归一化 .....	42
5.1.3 坐标系比例归一化 .....	43
5.1.4 坐标系方向归一化 .....	43
5.1.5 三维模型各向同性变换 .....	44
5.2 三维模型转化为二维深度图 .....	44
5.2.1 基本思路 .....	45
5.2.2 深度图像生成 .....	45
5.3 轮廓特征抽取 .....	46
5.3.1 边缘提取 .....	46
5.3.2 轮廓跟踪 .....	48
5.4 相似性计算 .....	49
5.5 本章小节 .....	51
<b>第六章 基于手绘草图的三维模型检索系统实现 .....</b>	<b>52</b>
6.1 引言 .....	52
6.2 系统功能结构和体系结构 .....	52
6.3 数据库设计 .....	54
6.4 关键技术实现 .....	56
6.4.1 轮廓抽取代码示例 .....	56



6.4.2 傅立叶变换代码示例 .....	58
6.5 系统运行示例 .....	59
6.6 本章小节 .....	64
<b>第七章 总结展望 .....</b>	<b>65</b>
7.1 本文总结 .....	65
7.2 进一步研究工作 .....	65
<b>参考文献 .....</b>	<b>67</b>
<b>致谢 .....</b>	<b>71</b>
<b>在学期间的研究成果及发表的学术论文 .....</b>	<b>72</b>

## 图目录

图 1.1 典型的三维模型检索系统的总体框架.....	5
图 2.1 基于手绘草图检索系统的框架.....	8
图 2.2 草图识别流程.....	9
图 2.3 草图特征提取流程.....	10
图 2.4 三维模型特征提取流程.....	11
图 2.5 相关反馈过程.....	17
图 3.1 冗余点删除效果.....	20
图 3.2 聚点消除效果.....	20
图 3.3 不能拟合的笔划.....	21
图 3.4 需要拟合的笔划.....	21
图 3.5 单笔画封闭情况处理.....	21
图 3.6 不同角度下飞机的轮廓.....	22
图 3.7 笔划采样点.....	23
图 3.8 轮廓提取效果.....	23
图 3.9 边缘平滑处理.....	24
图 3.10 基本图元结构.....	25
图 3.11 多基本图元识别流程.....	26
图 3.12 特征面积示意图.....	26
图 4.1 等点采样效果.....	28
图 4.2 空圆特性 .....	32
图 4.3 最大化最小内角特性.....	32
图 4.4 局部优化 .....	32
图 4.5 两种特殊的情况.....	36
图 4.6 能量函数 $E(x)$ .....	39
图 4.7 傅立叶降维效果.....	39
图 4.8 狗轮廓特征向量.....	40
图 4.9 背部加宽图 and 原图特征对比.....	40
图 4.10 尾巴加长图 and 原图对比.....	40
图 4.11 尾巴剪除图 and 原图对比.....	40

图 4.12 封闭边界的多边形近似分裂算法.....	42
图 4.13 封闭边界的多边形近似合并算法.....	42
图 5.1 坐标平面的切分.....	47
图 5.2 投影图像 .....	47
图 5.3 坐标面切分及投影图像.....	47
图 5.4 边缘提取前后对比.....	50
图 5.5 原始图像和轮廓跟踪图像比较.....	50
图 5.6 相似性计算流程.....	51
图 5.7 多特征检索结果比较.....	52
图 6.1 系统功能结构.....	53
图 6.2 系统结构 .....	55
图 6.3 数据库设计.....	56
图 6.4 用户绘图界面.....	61
图 6.5 读入草图 .....	62
图 6.6 傅立叶变换飞机一.....	62
图 6.7 傅立叶变换飞机二.....	63
图 6.8 飞机一和飞机二特征向量比较.....	63
图 6.9 检索飞机输入示例.....	64
图 6.10 检索结果显示.....	64

# 第一章 绪论

## 1.1 研究背景

### 1.1.1 三维模型建模技术

随着三维数据获取技术、三维图形建模方法及图形硬件技术的发展,已经产生了越来越多的物体的三维模型数据库,如虚拟现实环境(virtual reality)、三维游戏(3D game)、工业实体 CAD 模型库等等。三维模型已成为继声音、图像和视频之后的第四种多媒体数据类型,日益发达的互联网和功能增强的计算机为人们对三维模型的共享和处理提供了条件,在不久的将来,人们可以象使用二维图像一样任意地使用三维模型。

三维模型不仅在数量方面快速增长,而且应用领域也越来越广泛。在工业产品设计、建筑设计、虚拟现实、计算机仿真、数字人、多媒体教学、分子生物学、教育、三维游戏和影视动画中都广泛地使用了三维模型,三维数据模型库也已经越来越普遍。在互联网上,也涌现出越来越多的三维模型库<sup>[1]</sup>。三维模型已经成为 MPEG-7<sup>[2~4]</sup>标准模型的一个重要研究部分。

### 1.1.2 手绘草图技术

草图是符号系统的一种特殊形式,具有语义、语法和模糊的特征,能满足设计人员纵向和横向的思维活动。通过手绘草图,设计师可以大致的画出草图的轮廓特征,然后调整思路,修改草图。草图是形状构造过程中的一个不确定形态,可以把设计师最为关注的信息保存在草图中。

草图设计的交互方式不再模拟桌面环境,而是模拟人在纸笔环境下进行交互。界面的隐喻(Metaphor)由桌面环境(Desktop)变为纸笔环境(Pen/Paper)。相对于基于键盘鼠标和显示器的人机交互方式,笔式手绘草图具有信息连续性、信息多维性等新的特征。这种方式能极大地提高人机交互的效率。也使得草图技术在室内设计、工艺设计、建筑设计、图像检索中得到广泛的应用。

手绘草图识别是根据人机交互过程中不断增长的草图信息,挖掘草图形状约束,领会用户输入意图,最终将自由的、不规则的草图识别成为规则几何形状。20 世纪 90 年代中期以来,人们对草图识别技术及其应用进行了大量的研究,取得了一大批成果。归纳起来,他们的研究成果主要包括四类:一是基于笔划标准分割的方法,如 Revankar 等<sup>[5]</sup>将手绘草图分割成几个部分,通过一些规则建立决策树,每个叶节点输出一种形状,但草图识别受到输入的随意性以及视觉偏差的影响,不可能将草图分割成标准的笔划;二是基于特征相似度计算方法,以组成图

形的图元类型和数目为特征计算相似度,这种方法速度极快,但是由于特征过于简单,检索结果不够精确;而 SRG 方法<sup>[6]</sup>则以图元间的简单关系为特征计算相似度,具有精确的识别结果,但时间开销过大;三是基于过滤器方法<sup>[7]</sup>,除圆和直线外,其它图形必须与 X 轴或者 Y 轴平行输入,这个要求对于在线草图输入过于苛刻,Fonseca 和 Jorge 及其研究小组对这种方法进行了改进<sup>[8]</sup>,但改进的方法无法区分彼此诸如五边形和六边形等很接近的图形,而且,其扩展性较差;四是基于机器学习方法<sup>[9~11]</sup>,神经网络、支撑向量机、隐马尔可夫模型等方法也已经应用于手绘草图识别,并具有识别率高的优点,但其对大样本数据的需求、训练时间长等缺点,限制了其推广范围。不同的草图识别技术具有各自的优点和适用的范围,随着草图识别技术的进一步发展,草图设计技术将会得到越来越广泛的应用。

### 1.1.3 三维模型检索现状

目前,基于关键字的三维模型检索技术已经广泛使用,例如百度等大型搜索引擎。使用关键字检索文本信息是非常有效的,但是使用关键字检索三维模型有一定的不足。首先必须对三维模型添加手工标注,定义可以反应三维模型的关键字。这种方法存在以下局限性:第一,关键字的标注往往依赖于标注者的主观因素,因此容易产生歧义性;第二,这种标注方式比较片面,难以充分描述复杂的三维模型;第三,三维模型的数量如此巨大,手工标注显然费时费力。

为了能够得到更理想的检索效果,基于内容的多媒体信息检索(Content-based Multimedia Retrieval)引起了研究人员的广泛兴趣<sup>[12~15]</sup>。基于内容的含义就是试图利用反映三维模型视觉特征的内容信息自动建立特征索引,达到检索三维模型的目的。基于内容的三维模型检索技术通过对视觉特征的相似性匹配来查找用户所要求的三维模型,这更贴近于人们在现实生活中靠直觉印象使用信息的方式<sup>[16]</sup>。三维模型检索包括专业三维模型检索和通用三维模型检索。目前研究主要集中在通用的三维模型检索。

从上个世纪八十年代开始,专业三维模型检索技术的研究已经取得一定进展,在一些领域得到实际的应用。比如在生物分子领域、机械零件领域、地形匹配领域等,并逐渐获得了较好的实用价值。专业三维模型检索中使用的三维模型除了几何属性外,还具有一些专业领域的特征信息,具有较强的专业性。

通用的三维模型检索系统发展相对缓慢,还没有形成商用系统,但是国内外一些大学已经开发出一些实验系统。比如美国的普林斯顿大学、卡耐基·梅隆大学、华盛顿大学、布朗大学以及 NEC 美国研究院、HP 以色列海法研究院等著名院校和研究机构都已开展基于内容的三维模型检索技术研究,并取得一定进展,相关的学术论文和著作也越来越多<sup>[17~20]</sup>。

通用领域三维模型检索系统的数据库包含了各个领域的三维模型,大部分模型的形状轮廓和拓扑结构都比较复杂,因此准确描述三维模型的特征比较复杂。三维模型的特征提取方法的

优劣直接决定了能否找到用户希望的目标模型，所以现有工作主要集中在特征提取方面。而三维模型特征的最主要特征是形状，因此模型的特征又被称为形状描述子，基于内容的三维模型检索也主要是基于三维模型的形状检索。

目前已经提出许多特征提取方法<sup>[21]</sup>，可以分为以下几类：

(1) 基于全局几何特征的方法，例如以三维模型的表面积、中心点、最小包围盒、主轴等全局的几何信息作为特征值。优点是，计算简单，节省存储空间，一定的全局几何信息具有几何不变性；缺点是，不同的形体可能具有同样的全局几何特征，对于小的扰动敏感。

(2) 基于统计形状分布的特征提取方法，例如对三维模型上任意两点的距离进行统计，分布曲线作为特征描述。优点是，具有几何不变性，计算不很复杂；但是特征描述效果也不够理想，只适合粗略分类。

(3) 基于几何结构分析的特征提取方法：例如 Vranic 等人提出的 Spherical Extent Function 方法<sup>[22]</sup>。优点是，特征描述效果较好；但是不具有几何不变性，需要在特征提取之前作标准化变换，特征值抗扰动性较差，但可以运用傅立叶变换的方法加以解决。

(4) 基于函数投影的特征提取方法<sup>[22]</sup>：通常利用球面调和函数(spherical harmonic function)的不变特性，代表性的方法是将三维空间分解为一系列具有不同半径的同心圆，然后分别为这些同心圆空间定义具有不同旋转不变性的频率函数，其集合构成三维模型的特征向量。这是目前公认比较理想的特征提取方法之一，常和其他的特征提取方法结合使用。

(5) 基于代数矩的分析方法，具有代表性的是 Zernike 矩方法。

(6) 基于骨架的提取方法：例如文献<sup>[23]</sup>采用了基于网格模型多分辨率 RG(Reeb Graph)的骨架提取方法。优点是鲁棒性较强，对于几何不同拓扑相似的模型也有较好的识别性能。

尽管基于内容的三维模型检索已经取得一定研究成果，但是还存在许多不足。通过三维模型示例进行检索虽然很容易操作，但是用户可能没有合适的三维模型用作示例。如果从系统中选择模型进行检索又包含了人工查询的工作，则增加了用户的检索时间。因此提供更加友好、智能、人性化的检索界面和检索方式非常重要。本文研究通过用户绘制草图，快速检索用户需要的三维模型。要寻找更好的更适合描述草图的特征仍然会是一个长期的任务。

#### 1.1.4 现有的三维模型检索系统简介

在三维模型检索系统研究领域，目前根据现有理论及算法已开发出一些原型系统，且大部分实现了基于 Web 平台的搜索。其中，加拿大国家研究院的 Paquet 等人开发的三维模型检索系统 Nefertiti 是第一个通用三维模型检索系统，所提取的特征主要包括形状和颜色两种。目前，比较著名的通用三维模型检索系统和搜索引擎主要有：

(1) 美国普林斯顿大学形状检索与分析 (Shape Retrieval and Analysis Group) 即实验室

开发的三维模型搜索引擎，提供了二维草图和三维草图的绘制界面，并拥有目前规模最大的模型数据库，并且包含多个领域的三维模型子库。

(2) 美国卡耐基·梅隆大学 AMP (Advanced Multimedia Processing) 实验室开发的三维模型检索系统，该系统同时结合了底层形状特征和语义特征进行检索，并提供了用户相关反馈的功能。

(3) 国莱比锡大学 CGIP (Computer Graphics and Image Processing) 实验室的 Saupe 和 Vranic 等人开发的基于 VRML 模型库的三维模型检索系统 CCCC (Content-based Classification of 3D models by Capturing spatial Characteristics)，该系统的坐标系归一化预处理算法较好，且用户界面友好。

(4) IBM 日本东京研究院的“三维 Web 环境”研究项目中的三维几何形状检索和分析系统，该系统的开发目的是对三维模型库进行知识管理。

(5) 希腊 ITI 学院信息处理实验室开发的基于 VRML 模型数据库的三维模型检索系统，除了可对用户指定模型在三维模型库中进行检索外，还提供了对库中模型进行两两比较的功能。

(6) 荷兰 Utrecht 大学的 GIVE (Geometry, Imaging and Virtual Environment) 实验室研发的三维形状搜索引擎，主要使用基于高斯曲率等形状特征实现基于模型形状的检索。

(7) 日本多媒体教育学院的 Suzuki 等人开发的基于 Web 的多边形模型检索系统 Ogden N，该系统同时使用模型的形状和颜色特征对 VRML 模型数据库实现检索，以建立三维的网络教学环境。

(8) 美国布朗大学的 LEMS (Laboratory for Engineering Man/Machine Systems) 实验室开发的三维模型检索系统，以二维图像为输入方式，在三维模型库中检索出相似的三维物体模型。

(9) 德国波恩大学 Klein 和 Novotni 等人开发的数字图书馆系统 RODA，该系统不对模型进行特征提取，而是通过计算不同模型之间的体积测定误差以及分叉轮廓进行检索。

(10) 中国台湾省台湾大学通讯与多媒体实验室研制的三维模型检索系统，该系统使用 MPEG-7 标准推荐的三维形状描述符和多视图描述符对模型特征进行提取，基于 Web 平台进行三维模型检索。

此外，还有一些用于专业领域的检索系统，如德国慕尼黑大学 Ankerst 等人研制的三维蛋白质分子模型检索系统以及苏格兰 Heriot-Watt 大学开发的基于 Web 的 CAD 模型搜索引擎 ShapeSifter 等。

## 1.2 三维模型检索系统的一般框架

基于内容检索的一种主要方法是将三维模型的形状映射到特征向量空间，得到一组特征向量，然后通过比较其特征向量的值来实现模型检索。其关键技术步骤有四个，分别是模型坐标

的标准化与预处理、特征提取与索引、相似性匹配以及检索结果输出。图 1.1 给出了一个典型的基于形状的三维模型检索系统的总体框架<sup>[24]</sup>。

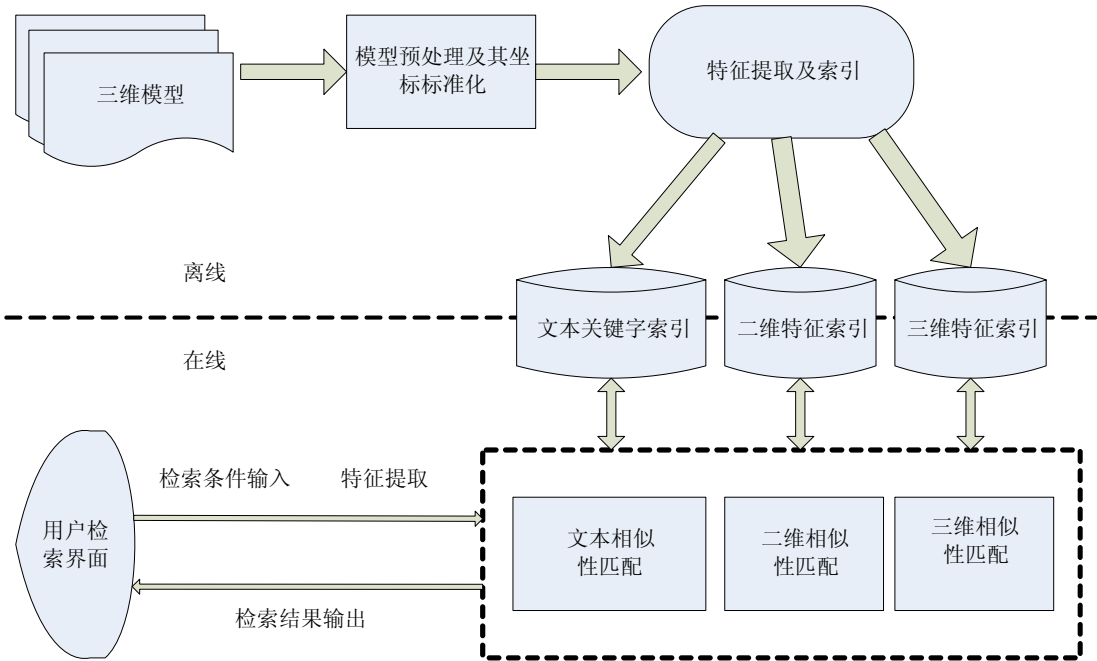


图 1.1 典型的三维模型检索系统的总体框架

### 1.3 研究的内容及其意义

三维数据模型库也已经越来越普遍。在互联网上，也涌现出越来越多的三维模型库。由于进行高逼真度的三维建模非常费时费力，如果能对已有的三维模型进行复用的话，将节省大量开支。比如在虚拟显示领域，一个具有高逼真度的三维模型建模比较费时费力，如果能够从模型库中的检索出的相似模型，然后经过简单的局部修改和调整，形成新的模型，将节约很多的时间和精力。再者，如果能在电子商务过程中实现高效的三维模型检索，用户就有可能检索到更为符合自身需求的目标产品，使得选购的过程更加的方便和快速。因此，如何有效地对三维模型及其场景进行浏览和检索，开始吸引了众多研究者的注意，并日益成为一个研究热点。

尽管基于关键词的检索技术已经取得重大成果，但是由于三维模型的复杂性以及用户表达的模糊性，使得基于关键词的三维模型检索方法已经无法满足设计和使用的要要。研究和开发三维模型搜索引擎帮助用户快速、准确地找到自己所需的三维模型，是一个迫切需要解决的问题。人们提出了诸多三维模型几何相似性比较算法，这些算法可以大致分为三类：第一类是基于轮廓形状的几何相似性比较算法；第二类是基于拓扑形状的三维模型几何相似性比较算法；第三类是基于视觉的三维模型几何相似性比较算法。

尽管基于内容的三维模型检索已经取得一定的进展，但是一般的基于内容的三维模型检索系统，需要输入三维模型，然后提取三维模型的特征，最后利用相似性匹配算法进行匹配。但



是需要用户提供三维模型是比较麻烦的，甚至用户对自己需要的模型也不是非常的清楚。本文通过手绘草图绘制界面，用户可以表达自己的检索需求。通过绘制草图快速检索三维模型具有重要的意义，同样面临巨大的挑战。

## 1.4 本文的工作及其组织结构

对于三维模型来说，形状是个复杂的概念，至今仍然没有统一的定义，三维模型可以通过多边形网格、体积、点云等表示。目前已有特征提取技术，各种特征提取方法有各自的优点和适用范围。本文主要研究可以应用手绘草图的特征提取技术，实现基于手绘草图的三维模型检索系统，具有重要的适用价值和研究意义。

第二章 介绍了基于手绘草图检索的总体框架，对系统的整个处理流程进行了描述，并对系统框架中各个模块涉及到的相关技术进行了概述。

第三章 提出了应用于三维模型检索的手绘草图的预处理技术和手绘草图识别技术和手绘草图特征提取技术。草图预处理包括冗余点消除、聚点消除、笔画拟合和闭包计算。草图经过预处理以后，草图被识别成外部笔画和内部笔画。其中，内部笔画被识别成基本图元。

第四章 研究了草图外围轮廓特征提取方法。轮廓描述的方法一般包括基于轮廓边缘的方法和基于区域的方法，本文使用基于轮廓边缘的方法。首先提取草图轮廓的粗略特征，然后使用四种方法描述草图的轮廓特征。

第五章 研究了三维模型特征提取技术、特征向量库生成方法、相似性计算。

第六章 介绍系统实现。首先系统的功能模块设计、系统结构模块的设计和数据库设计。然后给出系统运行界面和部分关键算法的源代码。

第七章 总结全文，并提出了下一步的工作重点。

## 第二章 基于手绘草图的三维模型检索系统总体框架

基于内容的三维模型检索的主要目标是如何在浩如烟海的三维模型数据库和网格中快速有效的搜索有用的、同主题模型。三维模型空间中一个简单的物体，即可以用多边形面片标示，也可以用点、线和体素标示，还可以由基于物体的几何操作和布尔操作表示。三维模型标示有自己的特殊性。首先，任意的三维曲面不存在类似于二维平面下弧长公式的直接参数化方法，因此无法对物体空间定义规则的、标准的类似于二维图像的采样方式。这种内在的不规则采样和三维表面存在任意的拓扑关系，使得经典的信号处理技术，如傅立叶变换，无法直接扩展应用到三维模型。其次，空间维数大小直接影响模型特征对应、参数匹配，分类其构造，优化路径搜索等问题，其复杂度与处理时间随之显著增加。可能的模型退化现象使得常用的几何形状特征，如面积、体积、曲率等难以计算。

另一方面而言，三维模型检索与场景语义无关，且不包含反射、阴影、遮挡、透射等效果，不需要考虑环境条件，极大地简化了匹配相似物体的过程。

因此，三维模型特征的选取，需要考虑以下三个方面。第一，特征要能够反映物体的内在基本特性。第二，三维模型在空间中的位置、朝向和大小是任意的，因此，所提取的特征应该与模型的平移、旋转、放缩等刚体变换无关。第三，即便对于退化的三维模型，特征也必须是确定的。噪声的存在和网格的细分操作不影响其稳定性和鲁棒性。

### 2.1 系统的体系结构

基于手绘草图的三维模型检索系统的总体框架如图 2.1 所示。用户输入的草图方法有两种：一种是通过鼠标或者手绘笔绘制自己的草图，然后检索三维模型；另一种是通过读入系统硬盘中的草图文件，然后检索三维模型。比如用户可以使用操作系统自带的画图程序，绘制草图后保存在系统硬盘里。

用户在线提交一幅手绘草图后，系统对草图的笔划进行预处理，然后提取草图的外围轮廓和内部基本图元。把识别的结果返回给用户，用户可以确认输入或者重新绘制自己需要的草图。为了对用户具有适应性，保存当前用户绘制的草图信息，获取用户的绘制草图的习惯，提高系统的用户适应性。对于系统中存在的三维模型，对其投影，得到投影轮廓，然后提取投影轮廓的特征，生成特征向量并保存在特征向量库里，用于三维模型检索。进行相似性计算是利用特征向量的距离计算查找用户需要的三维模型，然后按照相似程度从高到低返回给用户。用户对系统检索的结果进行评价，调整不同特征向量的权重，重新进行相似性计算。用户对结果的评价只包含相关和不相关，这样简化了相关反馈的处理。

图 2.1 中三个虚线边框指的是本文研究的重点部分：第一是草图识别，捕捉用户绘制的意图；第二是对识别的草图进行特征提取，方便检索用户需要的三维模型；第三是对系统模型库进行特征提取，生成特征向量库，用于相似性计算。

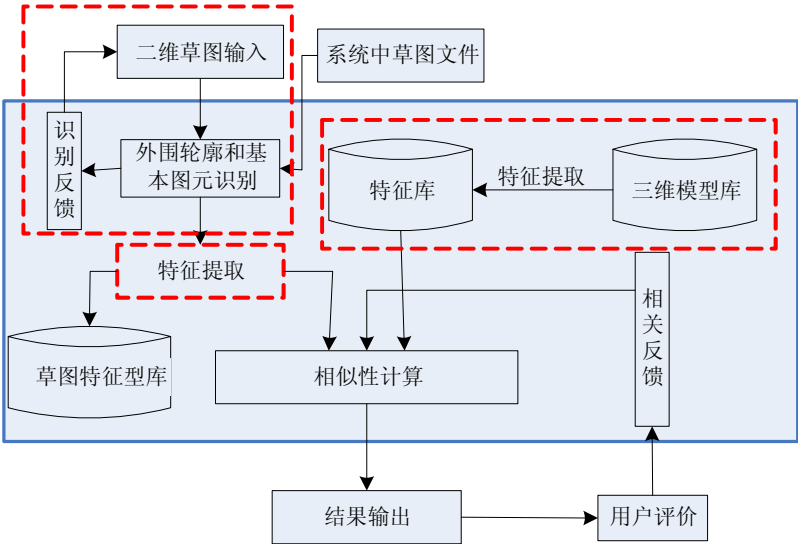


图 2.1 基于手绘草图检索系统的框架

## 2.2 系统处理流程

草图输入是用户对检索模型的表达，是系统处理的输入数据。草图输入是二维接口，用户可以通过系统运行的界面绘制草图，也可以通过文件读取方式获取保存电脑硬盘上边的草图。需要对用户输入的草图进行准确的识别，才能提取用户的输入意图。由于不同用户绘制的习惯不同，并使系统具有用户适应性，需要记录用户历史的绘制信息。草图识别的结果不一定正确，这需要和用户交互，确认识别。通过用户反馈，进一步识别草图。系统的识别经过用户确认后，提取其特征向量，进行三维模型检索。

### 2.2.1 草图识别数据流程

草图识别数据流程如图 2.2 所示。用户通过系统绘制草图界面输入草图，系统对输入的点序列进行相关计算。相关计算包括顶点的绘制方向、速度、曲率等。一般用户在绘制图形的时候，拐点处的速度和曲率和其它点处的速度有一定程度的不同，拐点处的速率偏小或者偏大，曲率偏大。而曲线的端点，速度偏小。根据相关性计算提取草图中的特征点，然后使用草图识别的基本原则识别草图。比如可以使用均方误差或者特征面，判断一条曲线是否可以用两个特征点的连线表示。一个封闭的轮廓线是否可以用最远两点的距离作为长轴，垂直平分线和轮廓的交点作为短轴的椭圆来近似。根据系统默认的识别准则识别当前草图，把识别结果反馈给用户。如果用户觉得识别结果不满意，可以重新调整参数，重新识别或者重新绘制草图。对于识

别满意的草图，用户可以保存草图的笔画，进一步细化草图。如果草图绘制结束，识别成功，那么把草图信息传入草图特征提取模块，提取草图的特征向量。

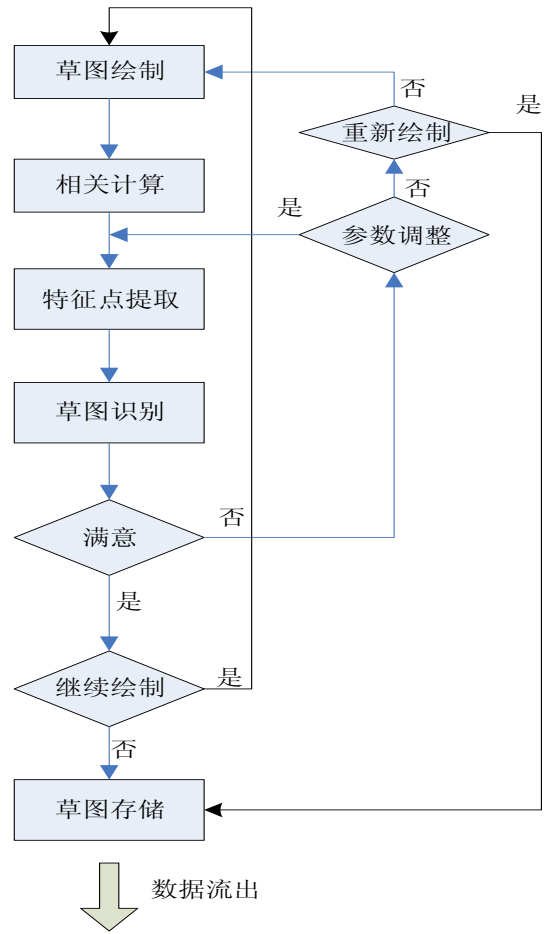


图 2.2 草图识别流程

### 2.2.2 草图特征提取流程

草图的特征包括外围轮廓特征和内部基本图元特征。对于三维模型来说，外围轮廓包含了模型的大量信息，是三维模型的重要特征。同样，内部的投影轮廓也包含一定的信息，可以提高检索的准确率。

草图的轮廓特征提取流程图如图 2.3 所示。首先对草图识别结果进行外围轮廓提取，剩下部分为内部笔画，然后用基本图元识别方法描述内部基本图元的特征。对于外围轮廓，一般是封闭的外围轮廓，因为三维模型的投影轮廓是封闭的。描述轮廓的方法一般有基于轮廓线的方法和基于区域的方法，本文主要研究基于轮廓线描述的方法。本文使用四种轮廓线的描述方法：等点采样方法、傅立叶变换方法、多边形描述方法和极图谱方法。

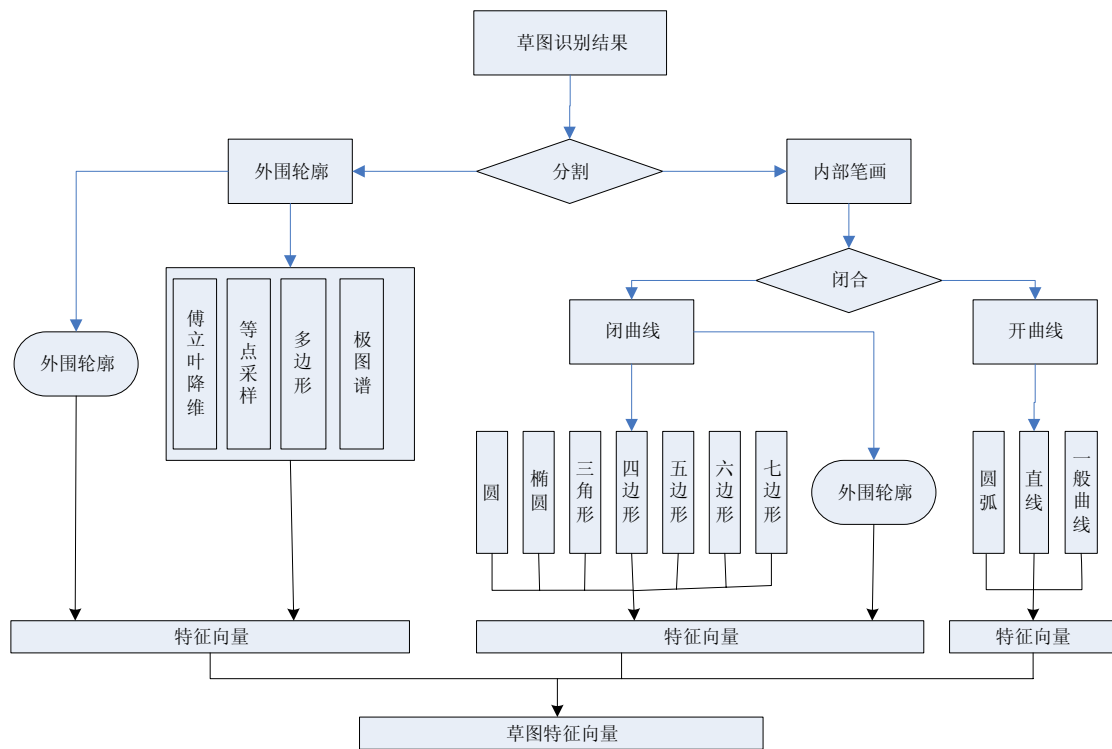


图 2.3 草图特征提取流程

内部基本图元特征，由于内部基本图元一般比较小，就是描述其大致特征，比如动物模型投影后，眼睛的投影轮廓比较小，可以用矩形或者椭圆去表示其形状。把封闭的轮廓分类：圆，椭圆，三角形，四边形，五边形，六边形，多边形。同样，多边形面积的 $4\pi$ 倍和边长平方的比值 LLS、任意三点组成的三角形面积和任意两点的距离等也是封闭图元的重要特征。基于手绘草图的三维模型检索主要是依靠外围轮廓特征，内部图元特征作为一种重要的补充。一般先对其外围轮廓匹配，当匹配结果可能符合要求的时候再进行内部基本图元匹配。

### 2.2.3 三维模型特征提取流程

三维模型特征提取流程如图 2.4 所示。由于检索的需要，算法要满足平移、旋转和比例缩放不变性。一般的基于内容的三维检索系统需要对三维模型进行预处理，进行归一化处理，然后再提取其特征。三维模型库经过处理以后，提取的特征满足平移、旋转、比例缩放不变性的要求。由于用户输入的草图是二维的，只有提取三维模型的二维特征才能进行相似性计算。为了得到模型不同角度的轮廓特征，选择多个投点投影三维模型，提取模型的投影轮廓，然后提取其特征。本文使用三视图正交投影，得到三维模型的 6 个投影图像对应 6 个外围轮廓图形。提取投影图形的特征后，存入系统特征向量库用于相似性计算，具体提取方法详见第五章。

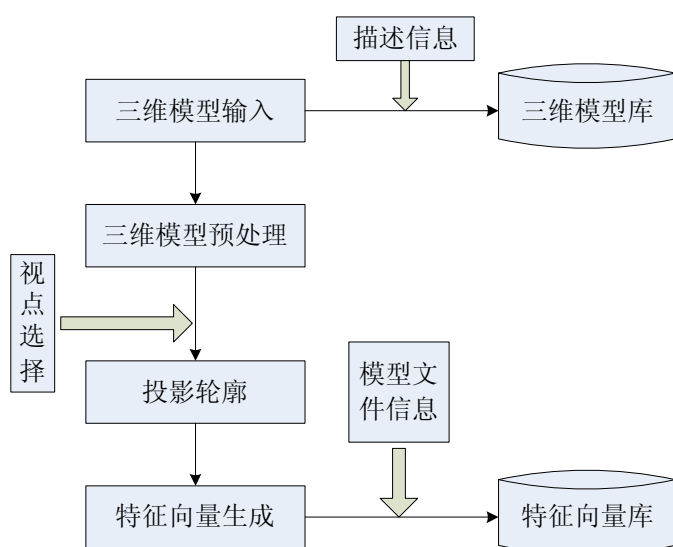


图 2.4 三维模型特征提取流程

## 2.3 关键技术

基于手绘草图的三维模型检索系统有五个关键技术，分别是三维模型检索中的草图识别、草图轮廓特征提取、三维模型特征向量库生成、相似性计算和相关反馈方法。本文主要研究前三个关键技术，同时对后两个关键技术进行必要的介绍。

### 2.3.1 三维模型检索系统中的草图识别

草图识别是系统的基础，只有准确的识别用户输入的草图，才能明白用户的检索意图，从而检索出用户需要的三维模型。用户绘制的草图是为了检索三维模型，绘制的是三维模型在某个投影面下的草图。三维模型的草图和一般的草图不一样，一般的草图可以用基本的图元表示，图元之间的基本关系复杂。三维模型的草图是三维模型在某个视点下的投影，三维模型的外围轮廓包含大量的信息。比如飞机、轮船、机枪等，仅从其外围轮廓就可以判断是飞机还是轮船或者机枪。而三维模型的内部轮廓不容易有草图绘制，表达容易产生歧义。只有具有明显意义的内部轮廓容易绘制，比如人的眼睛，机械的轴，轮船的客舱。所以草图的内部的轮廓一般比较简单，可以用基本的形状如圆、椭圆、三角形、四边形等基本图形表示。本文研究适合于三维模型检索的草图识别方法，和一般的草图识别方法有所不同。

用户在绘制草图的时候，思维是连续的，所绘制的草图也是连贯的。用户应当可以自由的绘制自己需要的草图，而不应当受到限制。同时用户绘制的草图，有时会有错误，多余的笔划，用户应当可以撤销刚刚绘制的笔划，而不是重新绘制整个草图。草图识别的同时应当可以保存当前绘制的笔划。

由于输入的随意性，本来相交的笔划可能不相交，闭合的曲线可能没有闭合或者在端点处有交叉。必需对草图进行预处理，规整用户输入的草图。然后识别草图，用户绘制的草图被识

别为外围轮廓和内部基本笔划。在绘制草图的时候，草图的采样点被保存下来，然后利用这些采样点计算其速度和曲率信息，提取一些具有代表性的点，把内部笔划识别为基本的图元。草图被表示为外围轮廓和内部基本图元信息。具体识别方法详见第三章。

### 2.3.2 草图外围轮廓特征提取

草图的外围轮廓特征是草图的最重要特征，主要利用外围轮廓特征检索需要的三维模型。草图的外围轮廓特征提取要满足平移、旋转、比例缩放不变性。提取草图的特征向量，大小不同的轮廓生成相同维数的特征向量有利于相似性计算，查找用户需要的三维模型。

由于模型库中的模型很多，所以特征向量库也很大，检索三维模型的时候逐一查找每一个轮廓的特征向量不仅消耗大量的时间，而且也没有必要。要对三维模型的投影轮廓建立索引，这样就不需要和特征向量库中的每一个特征向量都进行比较。本文使用紧凑度 LLS 建立索引，同时投影轮廓线上任意两点间的距离  $D$ ，任意三点所形成的三角形面积  $S$  也是重要的特征。检索三维模型的时候首先通过草图的 LLS、 $D$  和  $S$  检索可能相似的投影轮廓。

要想检索出用户需要的三维模型，需要提取轮廓精确的特征。本文使用四种基于轮廓线的方法提取草图的轮廓特征，用于相似性计算。具体外围轮廓的提取方法详见第四章。

### 2.3.3 系统特征向量库生成

查找三维模型是通过计算手绘草图的特征向量和三维模型之间的距离来实现的，不可能每次查找三维模型的时候都提取三维模型的特征然后进行相似性计算，一般需要一次性读取三维模型后提取其特征向量，然后存入系统特征向量库。

由于三维模型是三维的，而手绘草图是二维的，必需把三维模型转化为二维的才能进行相似性比较。用户绘制的草图是三维模型在某个视点下的投影轮廓，需要选择视点获得三维模型的投影轮廓。

选择的视点越多，得到的投影轮廓越丰富，系统特征向量库就越大。本文使用三视图正交投影的方法获取三维模型的投影轮廓。一般三维模型为了获取平移、旋转和比例缩放的不变性，需要对三维模型预处理。具体的三维模型特征向量库计算方法详见第五章。

### 2.3.4 相似性计算方法

相似性计算就是利用提取的特征向检索用户需要的三维模型。相似性计算就转化为特征向量距离计算问题。特征向量生成的方法不同，特征向量的距离计算方法也不相同。由于草图的特征向量由多个部分组成，草图被分割为外围轮廓和内部基本图元两个部分，而提取外围轮廓的特征使用四种方法。本文使用的特征向量距离计算方法有如下三种。

(1) Minkowski-form 距 ( $L_p$ ) 如式 2-1 所示，当  $p=2$  时，代表 EU 距离（或欧氏距离）。

可以用来计算一般特征向量的距离，距离越小，代表多维空间的距离越近。

$$D(X, Y) = \left( \sum_i |X_i - Y_i|^p \right)^{1/p} \quad 2-1$$

(2)  $\chi^2$  统计距如式 2-2 所示。

$$D(X, Y) = \sum_i \frac{(X_i - Y_i)^2}{X_i + Y_i} \quad 2-2$$

(3) 动态时间规整距  $d_{tw}$  [25] 定义如式 2-3 所示。对于长度为 N 的两个向量， $Q = \langle Q_0, Q_1, \dots, Q_{N-1} \rangle$  和  $T = \langle T_0, T_1, \dots, T_{N-1} \rangle$ 。Tail(Q) 定义为向量 Q 去掉第一个元素的向量即  $\text{Tail}(Q) = \langle Q_1, \dots, Q_{N-1} \rangle$ 。

$$d_{tw}(\langle \rangle, \langle \rangle) = 0, \quad d_{tw}(Q, \langle \rangle) = d_{tw}(\langle \rangle, T) = \infty$$

$$d_{tw}(Q, T) = \sqrt{(q_0, t_0)^2 + \min \begin{cases} d_{tw}(Q, \text{Tail}(T)) \\ d_{tw}(\text{Tail}(Q), T) \\ d_{tw}(\text{Tail}(Q), \text{Tail}(T)) \end{cases}} \quad 2-3$$

不同的相似性计算方法有不同的检索结果，计算量也不同。同样，不同的特征提取方法所生成的特征向量计算方法也不一样。比如等点采样形成的极心距特征向量和基于傅立叶降维生成的特征向量计算方法不一样，对于傅立叶降维生成的特征向量可以使用时欧氏距离和动态时间规整距离。而等点采样生成的极心距不具有旋转不变性，计算特征向量距离的时候必须进行处理保证其具有旋转不变性。

## 2.3.5 相关反馈

### 2.3.5.1 相关反馈概念

相关反馈是信息检索领域一直备受关注的研究方向，它是一种交互技术，旨在提供检索系统中基于用户评价相关记录的查询表达性能。相关反馈技术的运用希望达到下面的目的。

(1) 查询扩展：可以对用户不够具体的初始查询进行细化、具体化。将已经检索的结果不断缩小范围，最终得到相关度最大的结果集。

(2) 查询修正：当用户的查询目的不是非常明确，或者对查询结果不满意时，系统可以根据用户对初次查询结果的相关性判断调整查询算法来改变记录在查询记录中的位置，从而改变检索结果。

相关反馈是信息检索领域中的经典方法[26]，它根据用户先前检索结果与需求相关性的反馈信息自动地调整已有的查询使之更好地吻合用户的需求[27]。

相关反馈技术的基本思想是在检索过程中允许用户对检索结果进行评价和标记，指出结果



中哪些是与查询信息相关的，哪些是不相关的，然后将用户标记的相关信息作为训练样本反馈给系统进行学习，以指导下一轮检索，从而使得检索结果更加符合用户的需要。典型的相关反馈方法，一方面是采用修改查询向量的方法使其向相关向量的中心移动来改进查询结果。另一方面，利用反馈信息修改距离公式中各分量的权值，突出查询向量中较为重要的分量。

其形式化的定义如下：定义一个映射  $j: \{1, 2, \dots, n\} \rightarrow \{0, 1, \dots, n-1\}, r \mapsto j(r)$  是使关系  $\rho(q, d_{j(1)}) \geq \rho(q, d_{j(2)}) \geq \dots \geq \rho(q, d_{j(n)})$  满足的一个一一映射函数。同时，设  $D_r(q) = \{d_{j(1)}, d_{j(2)}, \dots, d_{j(n)}\}$  是查询信息结果中的前  $r$  个对象集合。 $D_r^{rel}(q)$  和  $D_r^{non}(q)$  分别是与查询相关和不相关的对象集合，且  $D_r^{rel}(q) + D_r^{non}(q) = D_r(q)$ 。对于固定的查询  $q$ ，可简写为  $D_r, D_r^{rel}, D_r^{non}$ 。

相关反馈的过程就是通过迭代从旧的查询结果以及相关对象和不相关对象推导出新的查询  $q'$ （如 2-4 式所示）和权矢量  $w'$ （如 2-5 式所示）。

$$q' = f(q, \{d_j' \mid d_j \in D_r^{rel}\}, \{d_j' \mid d_j \in D_r^{non}\}) \quad 2-4$$

$$w' = g(w, \{d_j' \mid d_j \in D_r^{rel}\}, \{d_j' \mid d_j \in D_r^{non}\}) \quad 2-5$$

从定义可以看出，相关反馈就是从用户与查询系统的实际交互过程中进行学习，发现并捕捉用户的实际查询意图，并以此修正系统的查询策略，从而得到与用户实际需求基于内容的图像检索技术研究系统与系统实现尽可能相吻合的查询结果。由于相关反馈可以实时地修改系统的查询策略，为检索系统增加了自适应功能。

### 2.3.5.2 相关反馈算法

由于相关反馈在三维模型检索中的重要性越来越受到人们的重视，在引入传统的基于文本的相关反馈方法的基础上，考虑到三维模型检索本身的特征，越来越多的学者致力于提出和引入新的相关反馈方法来提高用户交互的效率。下边从经典的三维模型检索相关反馈开始，介绍常见的各种相关反馈算法。根据系统对用户的不同反映途径，相关反馈算法可以分为基于参数调整的方法包括基于修改查询矢量或相似性度量权重的方法、基于优化的方法、基于人工智能学习的方法和带有记忆功能的相关反馈方法等等。

#### (1) 基于修改查询矢量或距离度量的相关反馈

在向量模型中，修改查询矢量最常用的方法是利用 Recchio 公式<sup>[28]</sup>改变用户检索向量（如 2-6 式所示）。

$$Q' = \alpha Q + \beta \left( \frac{1}{N_R^{rel}} \sum_{i \in D_R^{rel}} D_i \right) - \gamma \left( \frac{1}{N_R^{non}} \sum_{i \in D_R^{non}} D_i \right) \quad 2-6$$

其中  $\alpha, \beta, \gamma$  是常数， $D_R^{rel}$  是用户给出的相关模型集合， $D_R^{non}$  是用户给出的不相关图像集

合， $N_R^{rel}$  和  $N_R^{non}$  分别是  $D_R^{rel}$  和  $D_R^{non}$  中模型的数量。从公式可以看出，公式的目的就是要使正反馈相关的特征在检索样本中增强，与负反馈相关的特征减弱。形象的说，检索样本的修正就是在特征空间中根据用户的反馈来移动检索点，使其更接近好的检索位置，而远离“坏”的。所谓好的检索位置就是在特征空间中能带来更多正确检索结果的检索点的位置。

修改查询度量的方法通常是通过调整低层各个特征、特征各维的权重来优化检索结果。其基本思路是对在相关反馈结果集中的正反馈特征空间中找出特征分布标准差大的特征维降低权重，反之则增加权重，来提高检索的效果。特征的权重更新公式如 2-7 式所示。

$$w^{(K+1)} = 1 / \delta^{(k)} \quad 2-7$$

其中可  $\delta_{i,j}^k$  表示时刻查询结果正反馈的特征的标准差， $w^{k+1}$  为 k+1 时刻该特征的权重。标准差表示随机变量在数学期望周围取值的分散程度的一个数字特征，取值越分散，值越大，代表此特征对查询结果的影响小取值越集中，值越小，代表特征对查询结果影响大。

### (2) 基于优化方法的相关反馈

MinReader<sup>[29]</sup>使用一般的欧式距离进行特征向量的相似性度量，并且通过求解如 2-8 式的最优化问题求出最优的相似性度量函数中的权值矩阵。

$$\begin{cases} \min \sum_{x_i \in G} v_i (x_i - q)^T W (x_i - q) \\ s.t. \quad \det(W) = 1 \end{cases} \quad 2-8$$

其中 G 是反馈正例集合，q 为“最优”检索样本，W 为加权欧式距离函数中的权矩阵。 $v_i$  为  $x_i$  对应的相关度。当反馈集个数大于等于特征向量的维数时，可以得到一般加权欧式距离的最优权矩阵 W。但是在实际检索中，特征向量的维数往往明显高于每一轮反馈中用户给出的反馈正例的数量，所以只有当检索正例累积到一定程度时，一般的欧式距离才能有效改进检索结果。

Rui 综合了 MinReader 和 MARS 的反馈模型，提出了一个统一的优化问题框架<sup>[30]</sup>，并把一个带约束的优化问题通过引入拉格朗日算子转化成一个不带约束的优化问题来进行求解（如 2-9 式所示）。

$$L = \pi^T \times \bar{d} - \lambda \left( \sum_{i=1}^l \frac{1}{u_i} - 1 \right) - \sum_{i=1}^l \lambda_i (\det(W_i) - 1) \quad 2-9$$

其中  $\pi^T$  为用户标记的训练样本的相关性级别的集合，d 为正反馈模型对应的特征向量与“理想查询向量”的加权欧式距离，向量 u 和矩阵 W 为两个不同特征层次上的特征权值向量和权值矩阵。

### (3) 调整模型数据库分类或类间关系的相关反馈

调整模型数据库分类或类间关系的主要思想是首先用聚类方法对模型数据库做初步分类，然后根据用户的反馈信息对这些初始的分类结果进行类的合并或拆分。这种策略中比较好的应

用是 Lee<sup>[31]</sup>给出的聚类方法，根据用户的反馈进行类的分裂和合并，随着类的不断分裂，那些符合用户感知的类就是最终得到的查询结果。

文献<sup>[32]</sup>中构造了一个语义网络，每个模型都以不同的关键字和权重加以描述。关键词用于表达模型的语义，每个模型可以有多个关键字描述，模型的语义越简单，则用以描述的关键字越简单，关键词的权重反映了该词对模型语义描述的准确程度，在此基础上实现语义相关反馈，然后将该算法与视觉特征相结合，实现两个层次上的相关反馈，构造出一个统一的相似性度量方程。但是该方法的前提是对搜集来的存储在数据库中的模型进行预先标注或是一些已经分好类的图像数据或是网络上带有 TAG 标记的模型，之后还要对语义网络进行训练，这就给系统和用户带来了不必要的麻烦和开销。

#### (4) 基于人工智能学习方法的相关反馈

基于学习的方法的基本思想是根据用户反馈的信息样本进行统计推断。从本质上认为模型检索系统是一个分类问题，将把交互的过程看作是一个训练过程，用户认为比较满意和不满意的模型可以分别看作正例训练数据和反例训练数据，而进一步查询的过程是一个分类问题。传统的机器学习的方法都可以借鉴到相关反馈中来。由于分类器可以把所有的例子分为两类相关组和非相关组，这样就可以把许多模式识别领域的方法借鉴到相关反馈中来，如线性分类器、NN 分类器、贝叶斯分类器、支持向量机等。

### 2.3.5.3 用户反馈信息

图 2.5 显示了相关反馈的全过程，用户建立一个初始查询并提交给检索系统，该系统产生一个结果集，用户检查这个结果集，并按结果的质量或相关性给系统提供一个评价，系统根据原始结果和用户的反馈，建立一个新的查询。用户可以用三种方法来提供反馈信息：

(1) 好坏判别。用户观察每个检索结果，确定这个结果对于他的需求来说是好还是坏。虽然用户可以在不同粒度下提供相关反馈，但大部分检索系统只支持二元的相关信息：结果要么是相关的要么是不相关的。这种情况下，系统认为几乎所有的结果都是不相关的（或是不确定的），因为用户只能标记少数他们认为相关的结果。这种二元选择可以推广到多级相关性或不相关性的判断。有些系统使用可变的的多级相关，以损失用户的方便性换取更加准确的检索结果<sup>[33]</sup>。研究实验表明用户在这种多级反馈方式中通常只能给出很少的反馈，而且多级反馈的灵活性牺牲了用户的方便性<sup>[34]</sup>。

(2) 等级排序。这种方法中，用户每次考虑结果集的一个子集，然后按其所认为的相关度对其排序。这样用户在某种程度上实施了检索系统的任务：对结果进行相关性排序。这种方法可以被当作多级相关方法的一种扩展，它也可以有多个反馈等级，从而避免多个结果具有同样相关等级的情况。等级排序方法为检索系统提供了非常好的反馈信息，但对用户来说任务依然十分繁重。

(3)内部反馈。在内部反馈机制中，检索系统以可视化形式为用户呈现其内部查询结构，让用户交互地对这个内部结构进行操作，以改进查询表达。有些文本搜索引擎使用了内部反馈技术<sup>[35]</sup>。对于内部反馈，用户必须具备一定的领域知识，对复杂的多媒体数据有一定的了解。

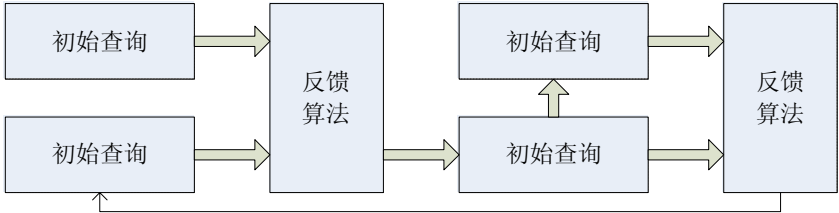


图 2.5 相关反馈过程

## 2.4 检索性能评估

检索策略性能评估的比较通用的两个准则是：效率 (efficiency) 和有效性 (effectivity)，分别代表检索的速度和查找与所查询的图像相似的图像的成功率。检索的性能用查准率 (Precision Ratio) 和查全率 (Recall Ratio) 来评估分别定义如下：

查准率=检索出的图像中相关图像的数目/检索出的图像数目

查全率=检索出的图像中相关图像的数目/图像库中所有相关图像的数目

查全率反映系统检索相关模型的能力，而查准率则反映系统拒绝无关模型的能力。试验表明，查全率和查准率之间存在这相反的相互依赖关系——如果提高系统的查全率，就会降低其查准率，反之亦然。

## 2.5 本章小结

基于内容的三维模型检索的主要目标是如何在浩如烟海的三维模型数据库快速有效的搜索有用的、同主题模型。如何有效的通过绘制二维草图，提取草图的特征，实现快速的获取用户需要的三维模型。本章提出了三维模型检索系统的一般框架，实现系统的三个关键点。关键点一，是草图识别模块，首先用户可以通过草图绘制界面，绘制自己需要的查询模型的草图；草图识别模块识别用户的输入，把识别结果反馈给用户，从而保证系统对草图的理解和用户的绘制意图相符合。关键点二，是草图特征提取模块，系统识别草图以后，只有提取草图的特征才能进行相似性计算；系统使用基于边界轮廓特征描述草图的外围轮廓特征，使用基本图元的方法描述草图的内部特征。关键点三，是三维模型特征提取模块，通过视图投影的方法，得到投影轮廓，然后提取特征，放入三维模型特征库。

草图经过预处理、草图识别、特征向量提取以后，可以利用计算草图特征向量和系统特征库中的距离进行相似性匹配。系统把检索的结果返回给用户，用户对反馈的结果进行评价。通过相关反馈系统，重新检索三维模型，把更符合用户需要的三维模型检索出来。

## 第三章 三维模型检索系统中的草图识别

草图识别是为了准确提取草图的特征，是基于手绘草图系统的一个重要模块。三维模型检索系统中的草图识别和一般的草图检索系统中使用的草图识别方法不一样。一般的草图检索系统，草图由基本图元组成，一个图元由多笔划或者单笔划绘制，而基本图元之间的关系相对复杂。而相对于三维模型检索系统中的草图识别，草图不能用简单的基本图元表示，因为草图的外围轮廓形状比较复杂，包含信息量大，同时内部基本图元之间的关系比较简单。三维模型一般是由在空间分布的点面组成，其对应的投影轮廓也是封闭的。三维模型的外围轮廓具有丰富的特征信息，一般不可以用简单的基本图形表示。外围轮廓应当用专门的轮廓描述方法去描述。而对于三维模型的内部轮廓，比如动物的眼睛的投影轮廓、汽车的天窗、机械的轴等，其投影轮廓相对简单，可以用简单的基本图元表示。所以本文识别草图，包括提取草图的外围轮廓特征和内部基本图元特征。草图识别过程为，首先对草图进行预处理，然后提取草图的外围轮廓，最后使用基本图元方法识别草图的内部笔划。

### 3.1 草图预处理

预处理是整个检索过程的基础，由于不同的绘制习惯，对于同一草图不同的用户绘制出来的草图可能会有大大的不同，甚至同一用户在不同时间绘制的草图也会有很大的差别。为了方便用户的使用，使用户的精力更多的集中于具体领域的工作，而不必在绘制草图的过程中花费更多的经历，系统对用户的绘制习惯不加任何的限制，甚至一个基本图元可以由多笔画输入。

通过鼠标移动事件采样记录笔画的点序列，并且为每个点记录时间值，作为草图的基本信息。由于设备的限制、用户绘制草图的习惯以及草图天生固有的随意性，输入的原始笔画中包含了大量的噪声，这为后面的图元识别带来一定的障碍。同时为了方便图元识别阶段的处理，在识别之前需要对草图进行处理以消除噪声。计算诸如输入点序列的方向、曲率等一系列的特征值，通过计算出的特征值消除噪声，具体实现将会在下文详细描述。

由于输入条件的限制和输入习惯的差异，用户输入的草图中会产生一些干扰笔划或信息，这些信息不利于机器的识别计算。一般来讲，噪声包括笔划冗余点和折点。为了便于对草图的拟合与识别，首先要消除这些噪声的影响。

#### 3.1.1 冗余点消除

受用户输入习惯的影响，采集草图中的样本点数往往很多。过多的采集样点会降低后续处理的性能。因此，如何在不明显改变图形原貌的同时，有效地消减样本点规模具有十分重要的

意义。实际上，输入草图是由大量折线段组成；有些折线段与由其相邻样本点构成的线段很相似，删除这些线段中的样本点将不会影响最终的识别效果。故在某种意义上说，这些点是冗余的。冗余点的定义可以通过冗余度进行刻画，其中冗余度为该点到相邻样本点构成线段的距离，若冗余度小于某给定阈值  $\varepsilon$ ，则该点可被判为冗余点。本文采用了由 Sklansky 和 Gonzalez<sup>[36]</sup> 提出的算法实现该思想。算法复杂度为  $O(n)$ ，其中  $n$  为样本点数。 $\varepsilon$  越小，保留点数目越多，草图笔划相对更平滑。效果如图 3.1 所示。

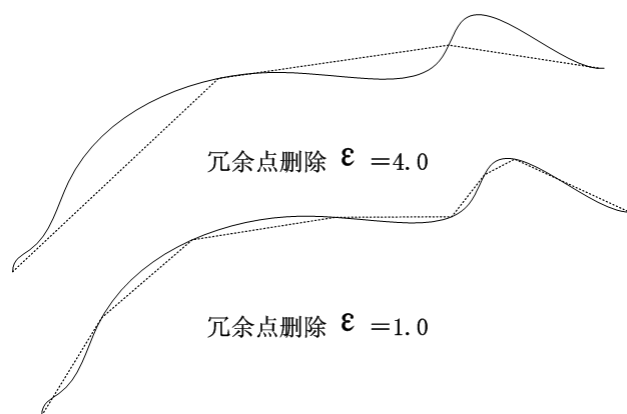


图 3.1 冗余点删除效果

### 3.1.2 聚点消除

聚点(Agglomerate Points)一般有两类<sup>[37]</sup>：钩子和圈如图 3.2 所示。这些点在冗余点消除中不会被去掉但是会影响后而图形的识别。考虑到转折处的点密度较大，本文定义长度为  $L$ ，样本点数为  $N$  的笔划的点密度为  $N/L$ ，笔划中任一段长度为  $l$  样本点数为  $n$  的折线的点密度为  $n/l$ ，如果相对密度  $Ln/Nl$  大于设定阈值，则认为该线段上的点均是聚点，消除方法是用重心代替。其算法复杂度为  $O(n^2)$ 。

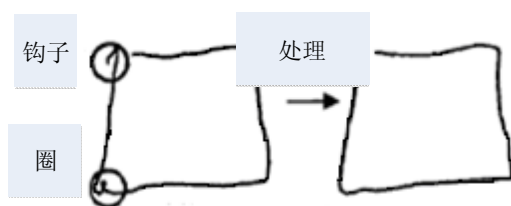


图 3.2 聚点消除效果

### 3.1.3 笔画的拟合

由于不同的绘制习惯，不同的用户绘制同一幅草图可能会大不相同，甚至同一个用户在不同时间绘制的草图也不相同；为了减少因绘制习惯的不同而带来的影响，可以根据不同笔划间的端点距离进行拟合。首先，检查一个笔划的端点，看其是否相邻（端点间的距离小于某个阈值）。如果两个端点不相邻（端点间的距离大于某个阈值），则这个笔划被称为开口笔划(en

stroke)。如果一个开口笔划的某个端点与另一个开口笔划的某个端点相邻，则这两个笔划要进行拟合。例如，在图 3.3 中，有个笔划的两个端点相邻，也就是说它不是开口笔划，所以这两个笔划无法拟合。而在图 3.4 中，3 个笔划都是开口笔划，而且一个笔划的端点与其它笔划的端点相邻，所以这 3 个笔划应该拟合为一个笔划。

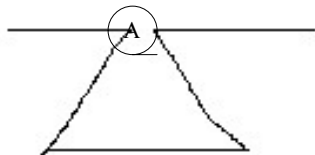


图 3.3 不能拟合的笔划

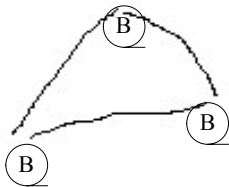


图 3.4 需要拟合的笔划

### 3.1.4 笔画闭合处理

在输入一个闭合曲线的时候，用户一般很难做到起点和终点重合。通常会出现图形端点分离或者交叉的情况，如图 3.5 所示。如果端点之间的距离/整个笔画的距离小于某一阈值，并且笔划自身没有相交，即图 3.5 (b) 所表示的情况，那么连接起点和终点；否则，此笔画是图 3.5 (a) 表示的情况，笔划自身相交，不需要把起点和终点连接。

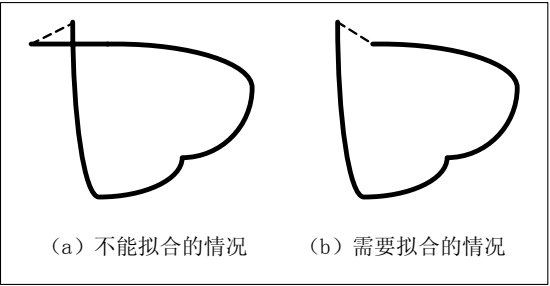


图 3.5 单笔画封闭情况处理

## 3.2 草图外围轮廓提取

形状是很多领域内的重要数据，比如在不同角度下边得到的飞机形状，如图 3.6 所示，总可以认出它是一个飞机而不是其它物体。草图的外围轮廓所表示的形状包含大量的信息，提取草图的轮廓并描述草图的形状信息具有重要的意义。

而对于人的模型，从正面，侧面，俯视看的时候得到各不相同的图像，从俯视图很难判断到底是人的俯视图还是其它物体的俯视图。而对于用户输入的草图，一般只是输入在某一个方向的投影图，所以要想找出用户想要的三维模型，必须与系统中模型多个视点下的轮廓图进行逐一比较。

对于形状的特征描述，需要提取轮廓的某些特征信息，生成特征向量，然后利用向量距离计算相似性。形状描述方法产生对应的形状描述向量，叫特征向量。形状描述的目标是用形状

描述的向量唯一刻画形状。一个形状描述方法应满足一定的性质，比如满足平移、旋转以及比例缩放不变性。因为这些变换按照其定义是不改变对象的形状，因此形状的描述也具有不变性。

下面详细介绍提取草图的外围轮廓特征：首先，对用户输入的草图进行预处理；然后，利用改进的八连通域的方向自适应跟踪算法提取草图的轮廓；最后，对提取的草图轮廓进行描述。



图 3.6 不同角度下飞机的轮廓

### 3.2.1 外围轮廓提取方法

对于用户提交的草图,用一种基于八连通域的方向自适应跟踪算法来对草图进行轮廓提取。以  $I(x, y)$  表示原始图像,  $C(x, y)$  表示轮廓的二值图像, 并定义当前方向  $d_i$ , 从正右方开始按照逆时针定义方向分别为  $0, 1, \dots, 7$ 。试验中算法具体实现: 选取图像最上边一行中最左边的一个点  $point\_c$  作为第一个点, 那么在以这个点为中心的  $3 \times 3$  领域内, 正上、左上、右上 3 个方向不存在其它点, 选取  $d_i=2$ , 并且按照逆时针方向寻找草图的外围轮廓。

具体算法描述如下。

(1) 初始化。将  $C$  清零, 令  $d_i=2$ , 方向数组  $DI$  赋值为  $DI = \{0, 1, 2, 3, 4, 5, 6, 7, 0, 1, 2, 3, 4, 5, 6, 7\}$ 。

(2) 按照从上到下, 从左到右逐行扫描原始图像  $I$ , 得到轮廓的起始点  $point\_c$ , 当前点  $point\_now$  初试化为  $point\_c$ 。

(3) 把当前点  $point\_now$  加入表示轮廓的二值图像  $C(x, y)$ 。搜索下一个点的顺序按照方向数组  $DI[d_i], DI[d_{i+1}], \dots, DI[d_{i+7}]$  的顺序, 设  $point\_next$  为  $point\_now$  点要查找的下一个相邻边界点。如果找到  $point\_now$  某一个方向上的点属于原始图像  $I(x, y)$ , 而且不等于初始点  $point\_c$ , 那么此点即为  $point\_next$ 。假设找到下一个点时的方向为  $DI[i]$ , 那么新的查找方向  $d_i$  为  $DI[i]$  相反方向的下一个方向。即  $d_i = DI[i] + 4 + 1 \bmod 8$ 。然后重新把  $point\_next$  的值赋值给  $point\_now$ 。



(4) 判断点  $point\_now$  是否与起点  $point\_c$  重合，是则退出，否则返回 (3)。

但是用户绘制的草图，系统采样点并不是连续的，系统采样点如图 3.7 所示。而基于八连通区域需要的二值图像是连续的像素点，所以需要系统采样点进行处理转化为可以通过基于八连通域的方向自适应跟踪算法提取的二值图。假设绘制草图的时候，鼠标移动的点序列被保存在数组  $Pen\_Point$  之中，那么可以利用数组  $Pen\_Point$  生成一个二值化数组  $I(x, y)$ ，即  $I(Pen\_Point[i].x, Pen\_Point[i].y)$  的值为  $true$ 。同时把  $Pen\_Point$  相邻两点之间的点也加入二值化数组  $I(x, y)$  中。图 3.8 显示手绘草图以及提取后的外围轮廓。



图 3.7 笔划采样点

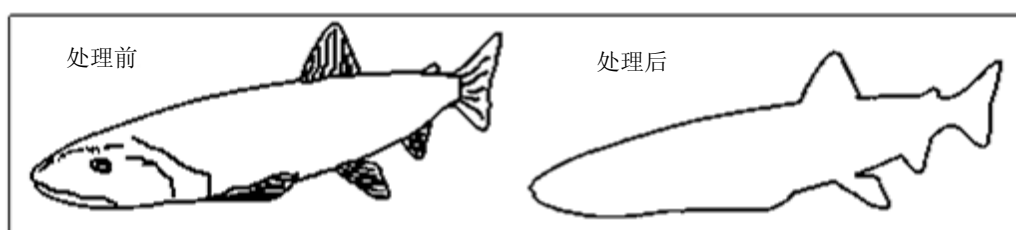


图 3.8 轮廓提取效果

### 3.2.2 平滑处理

由于手绘草图自身的特点，难免有两笔画交叉的地方，存在多余的非封闭的曲线段。对用户输入的草图进行必要的预处理，可以得到比较光滑的外围轮廓。边缘平滑处理效果如图 3.9 所示。经过平滑处理，消除草图中多余的非封闭的曲线段，得到轮廓比较平滑的草图。

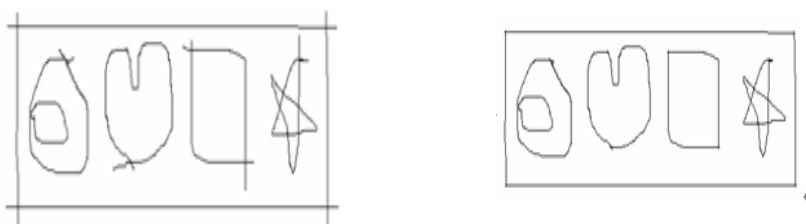


图 3.9 边缘平滑处理

消除非封闭曲线段算法描述：

(1) 按照行方向扫描图片，如果发现一个点属于曲线端点，那么转入 (2)，如果扫描完整个图片仍然没有曲线端点，那么退出。

(2) 找到曲线的端点，那么消除此端点，并判断此端点相邻的一点是不是曲线端点。如果是，那么继续消除此点并判断下一个相邻点。如果不是，那么转入 (1)。

曲线端点判断方法：

在一个点的  $3 \times 3$  的领域内。如果在这个点的八个方向上没有其它点存在说明是孤立点，属于曲线端点；如果在这八个方向中只有一个方向上有点存在那么是曲线端点；如果在这八个方向有三个以及三个以上方向上有点存在那么是非曲线端点；如果在八个方向中有两个方向上有点存在，这个两个方向相邻则是曲线端点，否则不是。

### 3.3 草图基本图元识别

#### 3.3.1 相关计算

为了识别草图的形状，要对草图上边的点进行相关计算。比如对于拐点，它的曲率会比较大。在绘制一条曲线的时候，端点的速度和中间点的速度不一样，可以通过提取速度特征判断直线一些特殊的点。同样，一个笔画的大致情况，到底是一条直线，还是折线，还是曲线的呢，这需要提取曲线的 NDDE 特征<sup>[38]</sup>，通过相关计算提取图形中具有代表性的点，然后识别用户绘制的草图。

草图笔划中每个点都是有序的，一个点  $P_i(x_i, y_i)$  的方向可以用矢量  $\overrightarrow{P_{i-1}P_i}$  的方向表示，但是不准确。本文使用 Sezgin 在文献<sup>[39]</sup>提出的方法：设置一个以待求点为中心的一个小窗口，求落在这个小窗口内的点序列的最小二乘拟合直线，然后用此直线的方向作为该点的方向值。速度是用户绘制图形时的一个重要特征，对于顶点探测也有十分重要的作用。本文用该点和其后续点之间的平均速度近似该点的即时速度，即速度表示为相邻两点的距离和相邻两点的时间的比值。

曲率表示该点出曲线的变化率，曲率越大表示曲线在此点处的弯曲程度越大。曲率用该点的方向角变化值和与该点相邻两点的距离的比值，其中该点的方向角变化值为  $2\pi$  与角  $\angle \overrightarrow{P_{i-1}P_i} \overrightarrow{P_iP_{i+1}}$  的差。NDDE 反映的是方向最大值点和最小值点之间笔画的长度占整个笔画长度的比例，对于如圆弧之类的曲线而言，最大方向点和最小方向点一般接近笔画端点，因此其 NDDE 接近 1。而折线的最大方向点（或者最小方向点）一般在笔画中间的某个顶点处，因此其 NDDE 一般小于 1。也就是说折线的 NDDE 比曲线的 NDDE 要小。NDDE 表示为最大方向和最小方向两点之间距离与整个笔画的长度的比值。

#### 3.3.2 多基本图元类型

一幅草图由多个笔划组成，对每一个基本笔划，本文提取语义信息，因此将它和一些给定的基本图元（直线、弧线和圆）相比，根据它和每种基本图元相似程度的大小将其归结为某种基本图元。将一个笔划看作一种基本图元后，一幅草图成为由基本图元构成的图元集合。由于草图的笔划构成不是唯一的，不同的用户可能绘制出不同笔划构成的同一幅草图，而草图的图

元构成是唯一的，所以利用图元构成可以获得草图的统一表示。在处理草图时，很容易得到笔划的几何信息，根据不同的几何特征识别笔划属于哪种基本图元：直线、弧线、曲线、圆、椭圆、多边形，如图 3.10 所示。

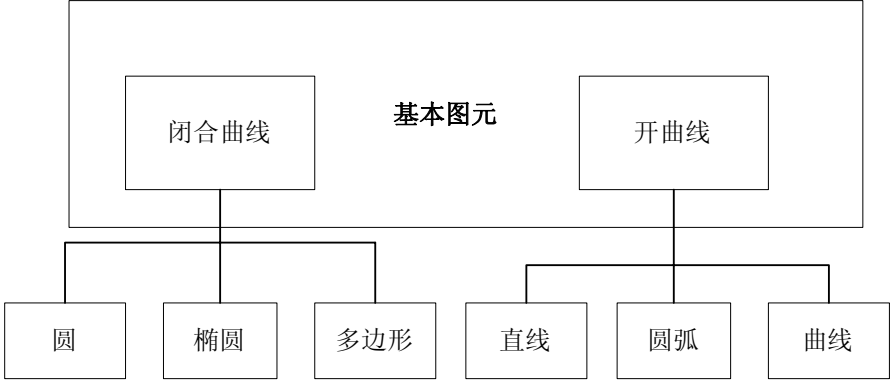


图 3.10 基本图元结构

### 3.3.3 基本图元识别流程

基本图元识别流程如图 3.11 所示。对于内部基本图元，首先使用 3.2.2 节提到的平滑处理方法，所有开曲线的基本图元会被消除，剩下的部分就是闭合曲线的图元。这里并不是真的把开曲线的基本图元消除，只是把这些笔划记录下来。

识别开曲线的基本图元。如果 NDDE 小于阈值，那么用最小二乘法拟合直线，然后使用特征面积除以笔划长度，如果小于阈值说明为直线。特征面积是 Yu 和 Cai 在文献<sup>[40]</sup>中提出来的，直线和曲线的特征面积如图 3.12 中阴影部分所示。如果不是直线，那么利用开曲线的两个端点和曲线中间一点三点确定一个圆。然后计算拟合误差，如果误差小于阈值说明为圆弧，否则为曲线。

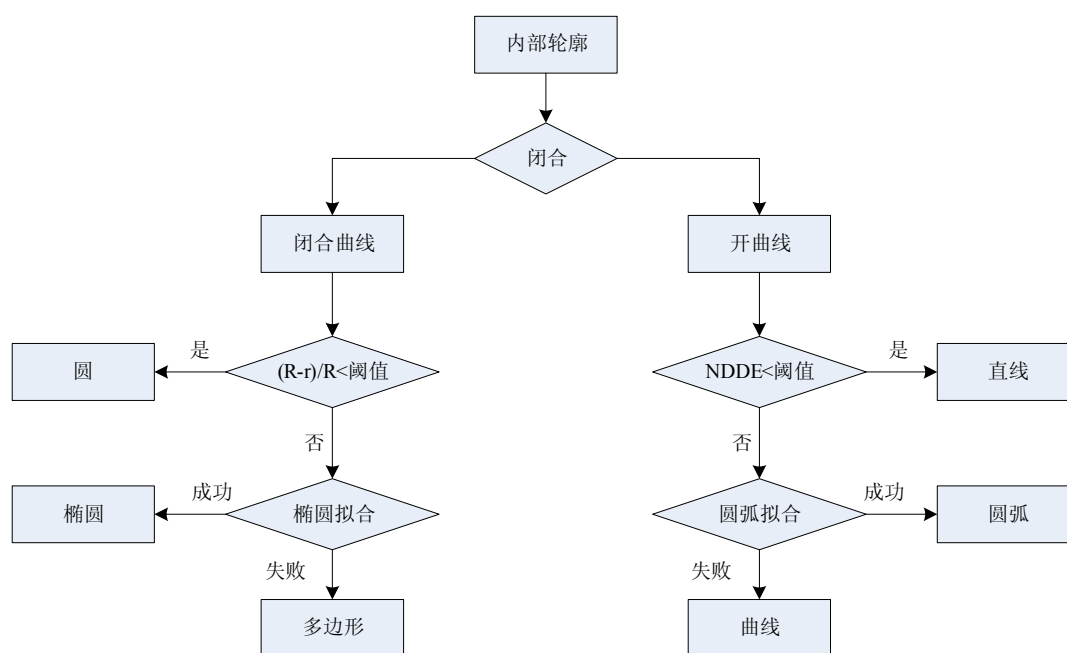
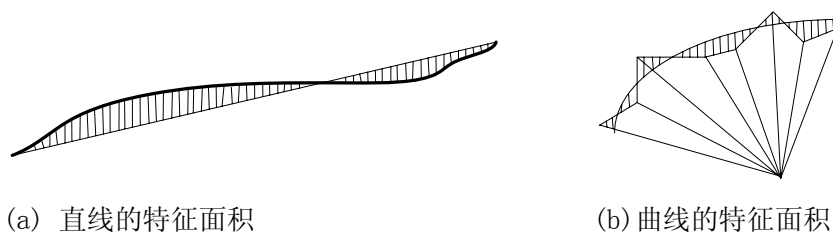


图 3.11 多基本图元识别流程



(a) 直线的特征面积

(b) 曲线的特征面积

图 3.12 特征面积示意图

对于闭合曲线的基本图元，首先判断是否为圆。首先使用基本图元上所有点的平均值作为基本图元的形心。然后计算基本图元上每一点到形心的距离，计算最大距离  $R$  和最后距离  $r$  的差  $\Delta R$ ，如果  $\Delta R / R$  小于阈值说明是圆。如果不是圆，那么用椭圆近似。计算基本图元上最大两点的距离  $a$ ，以这两点的中点作为椭圆的中心，这两点作为椭圆的长轴上两个端点。椭圆的短轴长度为长轴中垂线和基本图元上两个交点之间的距离。椭圆拟合误差小于阈值说明基本图元为椭圆，否则用多边形表示。多边形可以分为三角形，四边形，五边形，六边形和一般多边形。首先利用曲率变化大的点，速度比较快的点和比较慢的点提取出来。提取可以代表形状的少量特征点。如果提取的点还是比较多，删除曲率最小的点，计算特征面积，特征面积小于阈值就删除此点，否则删除失败。对于基本图元上边的少量的特征点，从这些点中选择三个点近似成三角形，如果特征面小于阈值那么拟合为三角形。如果三角形拟合失败，再拟合成四边形，五边形，六边形。如果上边拟合都失败代表绘制的是一般多边形，描述此多边形的特征量为  $VexNum$ （多边形的点数）和  $LLS$ （轮廓面积的  $4\pi$  倍和边长平方的比值）。

### 3.4 本章小节

本章主要介绍草图识别模块，提取草图的特征。由于用户绘制的草图具有随意性，用户的输入需要经过调整，噪声消除等处理，才能正确识别草图。首先经过草图预处理，消除冗余点、消除聚点、拟合笔画和闭合运算，识别用户输入的单笔画和多笔画。然后识别草图的外围轮廓和内部基本图元。本文使用基于八连通域的方向自适应跟踪算法来对草图进行轮廓提取，然后对提取的草图进行平滑处理，消除多余的线端。对于草图的内部笔画，使用基本图元方法进行描述。基本笔画包括圆、椭圆、多边形、直线、曲线和圆弧。通过草图预处理和草图识别，把草图表示为外围轮廓和内部基本图元。

## 第四章 草图外围轮廓特征提取

草图经过识别后，需要提取其特征才能应用于三维模型检索。本章主要介绍提取草图的外围轮廓特征。草图的外围轮廓特征包括多个方面，例如轮廓的长度  $L$ 、轮廓的面积  $S$ 、轮廓上任意两点之间的距离、任意三点所构成的三角形的面积、紧凑度等都在某个方面描述了轮廓的特征。以上几种特征只是粗略的描述轮廓的特征，为了能检索三维模型，必须使用更加精确的描述方法。外围轮廓特征的描述方法一般有基于区域的方法和基于轮廓线的方法，本文使用四种基于轮廓线的方法提取草图的外围轮廓特征。

### 4.1 基于等点采样的轮廓特征

草图的外围轮廓是由一系列连续的  $N$  个点组成的，等点分割就是等间隔选取  $M$  个点近似代表原来的轮廓，一般选择的  $M$  个点远小于  $N$ 。按照顺序把这  $M$  点连接起来，可以构造与草图轮廓相似的多边形近似，用多边形的顶点到形状的几何中心的距离及其与这个顶点相邻的两个顶点的距离构成的 3 个距离串来描述形状。这样描述形状具有计算简单、满足唯一性的优点，同时通过 3 个距离串对形状的全局特征和局部特征都进行了很好的刻画。

等点分割后得到的多边形边数与轮廓的长度无关，只要把轮廓等分为  $M$  个段，就可以得到  $M$  边形。不同的轮廓形状可以用相同个数的点来描述其形状，有利于特征提取和相似性计算。对鱼轮廓等点采样效果如图 4.1 所示。

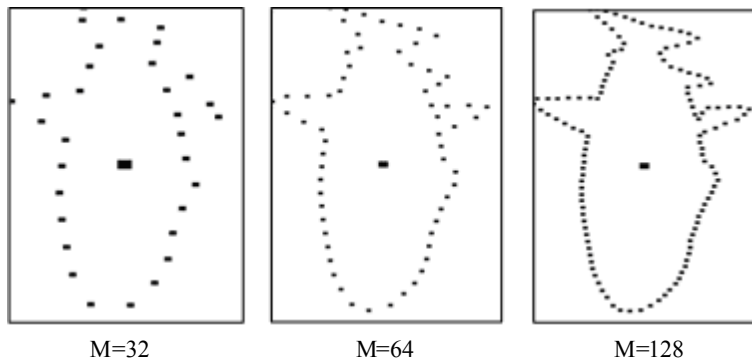


图 4.1 等点采样效果

#### 4.1.1 等点采样方法描述

按照第三章介绍的方法提取边界轮廓，然后逆时针遍历边界轮廓上的每一个点，按照遍历顺序把点存放在集合  $ARR\_POINT$  中。有序集  $ARR\_POINT$  定义如 4-1 式所示，其中  $u_i = (x_i, y_i)$ 。一般的情况是  $N$  比较大，而选取具有代表性的  $M$  个点近似描述轮廓。如图 4.1 所示，鱼的轮廓取点情况分别为 32、64 和 128，可以看出轮廓选取的点数越多，图形描述的越清楚，选择点数

较少只能大概描述其轮廓。当然当选取的点数越是多的时候，相似性计算时候对应的计算量也会扩大，所以试验中选择 64 个点。定义等点采样的宽度 Point\_w 为  $N/M$ ，其中  $N$  为边界点数， $M$  为选取的点数。 $N$  不一定可以被  $M$  整除，Point\_m 表示  $N \bmod M$ 。集合 EDG\_POINT 表示从有序集  $N$  个点中选出  $M$  个点的几集合，其定义如 4-2 式所示。集合 ARR\_POINT 和集合 EDG\_POINT 元素之间对应关系如 4-3 式所示。

$$\text{ARR\_POINT} = \{ u_1, u_2, \dots, u_i, u_{i+1}, \dots, u_N \} \quad 4-1$$

$$\text{EDG\_POINT} = \{ p_1, p_2, \dots, p_i, p_{i+1}, \dots, p_M \} \quad 4-2$$

$$p_i = \begin{cases} u_{1+point\_w*(i-1)} & i \in (1, point\_m) \\ u_{1+point\_w*(point\_m-1)+(point\_m-1)*(i-point\_m)} & i \in (point\_m+1, M) \end{cases} \quad 4-3$$

#### 4.1.2 特征提取匹配方法描述

对于集合 EDG\_POINT 中点组成的多边形，形心  $(x_0, y_0)$  纵坐标和横坐标定义如 4-4 式和 4-5 所示。 $d'_i$  为点  $(x_i, y_i)$  到形心  $(x_0, y_0)$  的距离，定义如 4-6 式所示。 $\lambda'_i$  为点  $(x_i, y_i)$  到上一顶点  $(x_{i-1}, y_{i-1})$  的距离，定义如 4-7 式所示。 $\gamma'_i$  为点  $(x_i, y_i)$  到下一顶点  $(x_{i+1}, y_{i+1})$  的距离，定义如 4-8 式所示。那么形状可以由三个距离串  $(d'_i, \lambda'_i, \gamma'_i)$  进行描述。大小不同的轮廓要进行归一化处理， $d_0$  如 4-9 式所示，经过归一化的距离串为  $(d'_i/d_0, \lambda'_i/d_0, \gamma'_i/d_0)$ ，记为  $(d_i, \lambda_i, \gamma_i)$ 。

$$x_0 = \frac{1}{N} \sum_{i=1}^N x_i \quad 4-4$$

$$y_0 = \frac{1}{N} \sum_{i=1}^N y_i \quad 4-5$$

$$d'_i = \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2} \quad 4-6$$

$$\lambda'_i = \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2} \quad 4-7$$

$$\gamma'_i = \sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2} \quad 4-8$$

$$d_0 = \frac{1}{M} \sum d'_i \quad 4-9$$

由于等点采样后近似的多边形的轮廓特征用三个距离串来刻画形状，对轮廓的全局特征和局部特征都具有一定的描述能力。形状的几何中心是形状的全局特征点，因而对形状的全局特

征具有一定的描述能力。轮廓线上的一段弧的弯曲程度是形状的局部特征，弧对应的弦长来描述这一特征。因为对于两个相同弧长的弧段，一般来说，对应的弦长短的弧段的弯曲的程度更大。而轮廓线被等弧长的分割成若干个弧段，所以可以用弦长，即多边形的顶点到其两个后继顶点的距离来描述轮廓线的局部弧段的弯曲程度。因而对形状的局部特征也进行了描述。

### 4.1.3 特征匹配

一个好的适宜于匹配的形状描述还应该满足平移、缩放和旋转不变性，因为这些几何变换并不改变目标的形状。将目标的轮廓线用基于等点分割的多边形来近似，而平移和旋转目标并不会使等分割点即多边形的顶点在轮廓线上发生移动。所以近似多边形的每一个顶点到其两个后继顶点的距离不会发生改变。又平移和旋转并不改变轮廓线上的每一个点到形状几何中心的距离。所以上文所使用的 3 个距离串描述满足平移和旋转不变性。当目标发生缩放时，近似多边形的每一个顶点到其两个后继顶点的距离以及到形状几何中心的距离都会发生相同倍数的缩放。上文使用轮廓上所有点到几何中心的平均距离  $d_0$  来标准化距离，所以 3 个距离串描述满足缩放不变性。

前面，用基于等点分割的多边形来近似轮廓线，用多边形的顶点到其两个后继顶点的距离以及到形状的几何中心的距离所构成的 3 个距离串来描述形状。那么，两个目标 A 和 B 的形状差异可以用它们的 3 个距离串的距离来度量。

设  $\bar{A}$  的 3 个距离串描述为  $d_i^A$ 、 $\lambda_i^A$  和  $\gamma_i^A$ ， $\bar{B}$  的 3 个距离串描述为  $d_i^B$ 、 $\lambda_i^B$  和  $\gamma_i^B$ ，特征向量  $\bar{A}$  定义分别如 4-10 式所示，特征向量  $\bar{B}$  定义如 4-11 式所示。因为目标的轮廓线是闭合的，所以得到的三个距离串是环形的。计算两个环形串的距离，首先要确定串的匹配的起始位置，所以本文用偏移量  $t$  来代表匹配的起始位置， $t$  的取值为 0, 1, ...,  $M-1$ ，对于  $t$  的每一个取值，完成一次串的匹配即计算两个串的距离，在  $M$  次向量距离计算中，取最小距离作为形状差异的度量。由于是一个环，所以有等式  $d_i = d_{i+M}$ 、 $\lambda_i = \lambda_{i+M}$  和  $\gamma_i = \gamma_{i+M}$  成立。 $\text{diff}_t$  代表向量  $\bar{A}$  的每一个向量元素和向量  $\bar{B}$  以第  $t$  个元素开始的每一个向量元素匹配的距离， $\text{diff}_{A,B}$  表示向量  $\bar{A}$  和向量  $\bar{B}$  的距离，求  $\text{diff}_{A,B}$  如 4-13 式所示， $\text{diff}_{A,B}$  越小，说明形状越相似。

以上计算形状差异度的公式可以得出，完成两个形状的匹配的时间复杂度为  $D(M^2)$ 。这里  $M$  是距离串的长度，即特征向量的维数。

$$\bar{A} = \{(d_i^A, \lambda_i^A, \gamma_i^A) | i = 1, 2, \dots, M\} \quad 4-10$$

$$\bar{B} = \{(d_i^B, \lambda_i^B, \gamma_i^B) | i = 1, 2, \dots, M\} \quad 4-11$$

$$\text{diff}_t = \sqrt{\sum_{i=1}^M (d_i^A - d_{i+t}^B)^2 + (\lambda_i^A - \lambda_{i+t}^B)^2 + (\gamma_i^A - \gamma_{i+t}^B)^2} \quad 4-12$$



$$diff_{A,B} = \min \{diff_t, t = 1, 2, \dots, M\}$$

4-13

## 4.2 基于极图谱的轮廓特征

### 4.2.1 Delaunay 图的关联图谱表示

#### (1) 三角剖分定义<sup>[41]</sup>

三角剖分：假设  $V$  是二维实数域上的有限点集，边  $e$  是由点集中的点作为端点构成的封闭线段， $E$  为  $e$  的集合。那么该点集  $V$  的一个三角剖分  $T=(V, E)$  是一个平面图  $G$ ，该平面图满足条件：

- 除了端点，平面图中的边不包含点集中的任何点。
- 没有相交边。
- 平面图中所有的面都是三角面，且所有三角面的合集是散点集  $V$  的凸包。

#### (2) Delaunay 三角剖分的定义

Delaunay 三角剖分是一种特殊的三角剖分。设  $E$  中的一条边  $e$ ，两个端点为  $a$  和  $b$ 。如果存在一个圆经过  $a$  点和  $b$  点并且圆内不含点集  $V$  中任何其他的点，那么这条边为 Delaunay 边。如果点集  $V$  的一个三角剖分  $T$  只包含 Delaunay 边，那么该三角剖分称为 Delaunay 三角剖分。

#### (3) Delaunay 三角剖分的准则

a、空圆特性：Delaunay 三角网是唯一的（任意四点不能共圆），在 Delaunay 三角形网中任一三角形的外接圆范围内不会有其它点存在，如图 4.2 所示。

b、最大化最小角特性：在散点集可能形成的三角剖分中，Delaunay 三角剖分所形成的三角形的最小角最大。从这个意义上讲，Delaunay 三角网是“最接近于规则化的”的三角网。具体的说是指在两个相邻的三角形构成凸四边形的对角线，在相互交换后，六个内角的最小角不再增大。如图 4.3 所示。

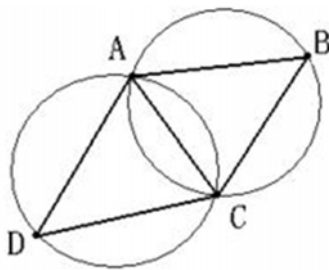


图 4.2 空圆特性

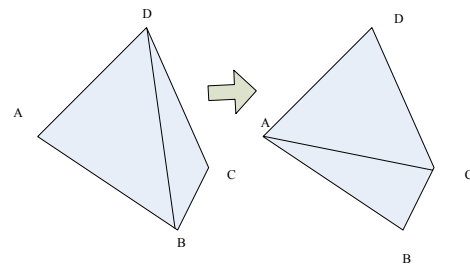


图 4.3 最大化最小内角特性

#### (4) Delaunay 三角剖分的特性

- 最接近：以最近临的三点形成三角形，且各线段(三角形的边)皆不相交。
- 唯一性：不论从区域何处开始构建，最终都将得到一致的结果。
- 最优性：任意两个相邻三角形形成的凸四边形的对角线如果可以互换的话，那么两个三

角形六个内角中最小的角度不会变大。

d. 最规则：如果将三角网中的每个三角形的最小角进行升序排列，则 Delaunay 三角网的排列得到的数值最大。

e. 区域性：新增、删除、移动某一个顶点时只会影响临近的三角形。

f. 具有凸多边形的外壳：三角网最外层的边界形成一个凸多边形的外壳。

(5) 局部最优化处理

理论上为了构造 Delaunay 三角网，Lawson 提出的局部优化过程 LOP，一般三角网经过 LOP 处理，即可确保成为 Delaunay 三角网，处理结果如图 4.4 所示。

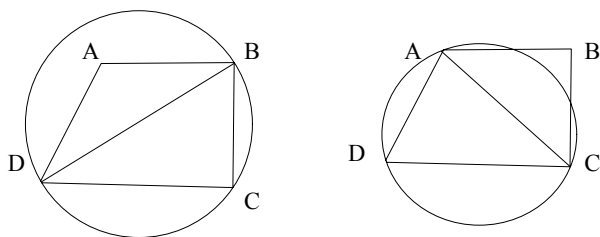


图 4.4 局部优化

其基本做法如下所示：

a. 将两个具有共同边的三角形合成一个多边形。

b. 以最大空圆准则作检查，看其第四个顶点是否在三角形的外接圆之内。

c. 如果在，修正对角线即将对角线对调，即完成局部优化过程的处理。

(6) 图的矩阵表示方法

表示图的存储方法有多种，本文使用矩阵来表示一个图的结构，可以使用的矩阵有邻接矩阵、加权矩阵和拉普拉斯矩阵。

对于图  $G = (V, E)$ ，邻接矩阵  $A_G(i, j)$  是一个大小为  $|V| \times |V|$  的矩阵，其中它的行和列分别都表示图  $G$  中的顶点，当某两个顶点  $(i, j)$  之间有边，则该矩阵中对应的元素值为 1，否则为 0。如公式所示如 4-14 式所示。

$$A_G(i, j) = \begin{cases} 1 & \text{if } (i, j) \in E \\ 0 & \text{otherwise} \end{cases} \quad 4-14$$

邻接矩阵是二值的，即只用和来描述图顶点的连接关系，可以表示图的基本结构。可以使用加权邻接矩阵，可以更准确的表示图形。对于图  $G = (V, E)$  加权邻接矩阵  $A'_G(i, j)$  是一个大小为  $|V| \times |V|$  的矩阵，其中它的行和列分别都表示图  $G$  中的顶点，当某两个顶点  $(i, j)$  之间有边，距离为  $d_{i,j}$ ，则该矩阵中对应的元素值为  $\exp(-d_{i,j} / \rho^2)$ ，否则为 0，如 4-15 式所示。

对于图  $G = (V, E)$   $A_G(i, j)$  是图  $G$  的邻接矩阵， $D_G(i, j)$  是图  $G$  的度对角矩阵， $D_G(i, j) = \text{diag}(d_1, d_2, \dots, d_{|V|})$ ，则矩阵  $L_G(i, j) = D_G(i, j) - A_G(i, j)$  就称为图的拉普拉斯矩

阵。

$$A'_G(i, j) = \begin{cases} \frac{d_{i,j}^2}{\rho^2} & \text{if } (i, j) \in E \\ 0 & \text{otherwise} \end{cases} \quad 4-15$$

邻接矩阵和拉普拉斯矩阵都是实对称矩阵，便于矩阵分解。邻接矩阵的行和列和反映顶点的度，拉普拉斯矩阵的对角线直接反映顶点的度。此外，拉普拉斯矩阵的特征值是非负值，更方便使用。加权邻接矩阵不但是对称矩阵，其矩阵元素值还反映出邻接顶点间的距离关系，更好地描述了关联图的结构。

#### (7) 关联图的谱分析

图的邻接矩阵虽然可以完整地表示图的结构，但是随着图的顶点数增加，图的邻接矩阵中的矩阵元素会按顶点数的平方倍数增加。显然用庞大的邻接矩阵来表示形状是不切实际的，更不能被用于基于形状的图像识别和检索。用关联图来表示形状的另外一个难题是比较两个较大的图的相似性，由于图的同构问题是一个 NP 问题，所以这也就制约了关联图方法的应用。关联图谱方法是近似解决该问题的一种有效方法，因为图的谱在很大程度上反映了图的结构。需要注意的是一个图的谱并不能唯一地确定一个图，例如文献<sup>[42]</sup>中指出“几乎所有的树都是同谱的”，因为树也是一种特殊的图。

图  $G = (V, E)$ ， $A_G(i, j)$  是图  $G$  的邻接矩阵，是一个实对称矩阵，它的特征值  $(\lambda_1, \lambda_2, \dots, \lambda_{|V|})$  为实数。不妨将这些特征值从大到小排列起来  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_{|V|}$ ，这些特征值不与顶点序号不同而改变，本文记  $\lambda_i(G) = \lambda_i(A_G) = \lambda_i$ ，其中  $i = 1, 2, \dots, |V|$ 。称  $Spec(G) = \{\lambda_1(G), \lambda_2(G), \dots, \lambda_{|V|}(G)\}$  为图  $G$  的谱半径。

图的谱理论<sup>[43,44]</sup>主要涉及图的邻接矩阵的谱和图的拉普拉斯矩阵的谱，是图论特（别是代数图论）和组合矩阵论共同关注的一个重要课题。其研究的主要途径是，通过图的矩阵（邻接矩阵或拉普拉斯矩阵）表示，建立图的拓扑结构（特别是图的各种不变量）和图的矩阵表示的置换相似不变量之间的联系，通过矩阵论，特别是非负矩阵理论和对称矩阵理论，和组合矩阵论中的经典结论用于图的拓扑结构的研究，同时也将图论的经典结论用于非负矩阵理论和组合矩阵论，以推动后者的理论研究。

关于图的邻接谱的研究最早源于量子化学研究领域，对非饱和碳氢化合物的一种近似处理产生了对应分子的图论模型<sup>[45]</sup>，其中图的特征值被用来表示特定电子的能量级。图谱方法在物理，机械工程，地理，社会科学等方面都到了广泛的应用，并且图谱自身的理论研究也迅速发展起来。

假定有  $N$  个关联图  $G_1, G_2, \dots, G_k, \dots, G_N$ ，对每个图  $G_k$  计算得到对应的邻接矩阵  $A_k$ 。首先

通过公式  $|A_k - \lambda_k^w I| = 0$  来计算邻接矩阵  $A_k$  的特征值  $\lambda_k^w$ ，求解方程  $A_k \varphi_k^w = \lambda_k^w \varphi_k^w$  得到特征向量  $\varphi_k^w$ ，其中  $w = (1, 2, \dots, |V_k|)$  是特征模索引。把得到的特征值按照从大到小的顺序排列  $|\lambda_k^1| > |\lambda_k^2| > \dots > |\lambda_k^{|V_k|}|$ 。并将这些特征向量  $\varphi_k^w$  也按照特征值的顺序排列成矩阵形式  $\Phi_k = (\varphi_k^1, \varphi_k^2, \dots, \varphi_k^{|V_k|})$ 。

当得到邻接矩阵的特征值和特征向量后，就可以将第  $k$  个图的邻接矩阵进行谱分解， $A_k = \sum_{w=1}^{|V_k|} \lambda_k^w \varphi_k^w (\varphi_k^w)^k$ 。如果假定  $A_k = \text{diag}(\lambda_k^1, \lambda_k^2, \dots, \lambda_k^{|V_k|})$  是  $A_k$  特征值组成的对角矩阵，则又可以写成  $A_k = \Phi_k A^k \Phi_k'$ 。模邻接矩阵定义为  $S_k^w = \varphi_k^w (\varphi_k^w)'$ 。

根据矩阵理论，一个矩阵的谱信息主要集中在它的较大的若干个特征值。因此，本文使用前面  $n$  个特征值对应的模特征矩阵  $\Phi_k = (\varphi_k^1, \varphi_k^2, \dots, \varphi_k^n)$ 。图的谱分析主要根据模特征矩阵和邻接矩阵的特征值来研究邻接矩阵的图谱特征，主要包括主分量特征值、特征模体积、特征模周界、常数、模间邻接矩阵和模间距离。

#### 4.2.2 极图谱特征提取

本文使用主分量特征值构成的向量作为 Delaunay 图的谱特征， $F_k = (|\lambda_k^1|, |\lambda_k^2|, \dots, |\lambda_k^n|)$ ，其中  $|\lambda_k^1| > |\lambda_k^2| > \dots > |\lambda_k^n|$ ，也就是图  $G_k$  的谱，并取绝对值。

基于极图谱的形状描述是本文提出了一种由物体形状轮廓点直接生成 delaunay 图的方法。首先对图像中的形状做初级规范化处理，计算形状的形心，然后计算所有物体形状边缘点到形心的距离。以形心作为极坐标系的极点建立极坐标系，根据距离最远的轮廓点重新归一化所有轮廓点的方向，通过极坐标系的极角来选取形状轮廓边缘的点作为图的顶点来构造 delaunay 图。最后计算每个 delaunay 图的图谱来代表该 delaunay 图的特征向量，用于形状聚类和检索。

$$\rho_i = \sqrt{(\Delta x^2 + \Delta y^2)} \quad 4-16$$

$$\theta_i = \begin{cases} -\arctan(\Delta x / \Delta y) & \text{if } \Delta x > 0 \text{ and } \Delta y \leq 0 \\ \pi / 2 & \text{if } \Delta x = 0 \text{ and } \Delta y < 0 \\ \arctan(\Delta x / \Delta y) + \pi & \text{if } \Delta x < 0 \text{ and } \Delta y \neq 0 \\ 3\pi / 2 & \text{if } \Delta x = 0 \text{ and } \Delta y \geq 0 \\ 2\pi - \arctan(\Delta x / \Delta y) & \text{if } \Delta x > 0 \text{ and } \Delta y > 0 \end{cases} \quad 4-17$$

##### (1) 形心

基于极图谱的形状描述的第一步是定位一个形状的形心。为了获得平移、旋转和缩放不变性，形状的几何中心被选为一幅图像的参考点。本文使用所有边缘点坐标的平均值来计算形心，如 4-5 式所示，其中  $N$  代表所有的形状轮廓点的个数。

##### (2) 形心角和形心距

所有形状边缘点到形心的距离称作形心距 ( $\rho_i$ )，如 4-16 式所示。其中  $\Delta x = x_i - x_0$

和  $\Delta y = y_i - y_0$  形心距的直线与水平的极轴方向的逆时针夹角被称作形心角  $\theta_i$ ，如 4-17 式所示。把极坐标系统的极点放在形心的位置，把坐标原点和形心进行统一。

### (3) 极图和极图谱

得到了物体形状在极坐标系统下的表示  $(\rho_i, \theta_i)$ 。为了获取物体形状的旋转不变性，将整个极坐标空间旋转，使得最大形心距  $(\rho_{\max})$  的方向与极轴重合，其他的形心角  $(\theta_i)$  需要根据最大形心距  $(\rho_{\max})$  的形心角  $(\theta_{\max})$  重新计算。

因为图形本身的复杂性限制和 Delaunay 图形本身的特点，不能选择所有的边缘点作为图的顶点来建立 Delaunay 图。设想这样来抽取形状的轮廓点：假设有许多射线从形心出发，这些射线将与物体形状产生相交点。本文就选取这些交点作为图的顶点。因为一个稍微复杂的物体形状很难用数学方程来描述，所以直接求出射线与形状的交点比较困难。

转换一下思考角度，前面已经计算出所有轮廓边缘点的形心角  $(\theta_i)$ ，因此就可以利用形心角来选择合适的轮廓边缘点来构造 Delaunay 图。例如，选择等间隔  $\Delta\theta$ ，显然可以得到物体形状在相对形心的  $2\pi/\Delta\theta$  个形心角方向的边缘点。当然， $\Delta\theta$  是可变的等间隔的大小，如  $\pi/8$ ， $\pi/16$ ， $\dots$ ， $\pi/180$ 。这样就可以得到不同密度的物体形状边缘的像素点，构造出来的图的大小也不一样，描述形状的准确程度也就不一样。容易发现，随着选择抽取角度  $\Delta$  的由大到小，将得到对物体形状由粗到细的多分辨率表示。

由于物体形状轮廓的复杂性，在实际提取形状边缘过程中，可能会遇到这样两种特殊情况一种情况是一些边缘点具有相同的形心角，这样就会在形心发出的射线上得到不止一个的边缘点，如何选取合适的边缘点来作为图的顶点。第二种情况是某些边缘点与形心十分接近，导致在该方向上抽取不到形状的边缘点。这两种情况如图 4.5 所示。

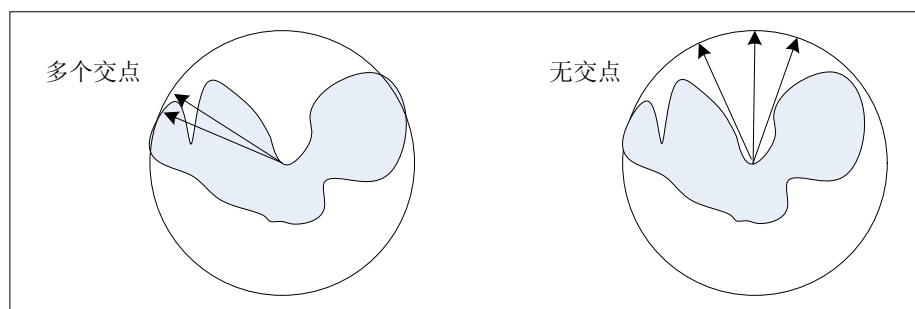


图 4.5 两种特殊的情况

在 Delaunay 图的比较和匹配问题上，顶点数相同的 Delaunay 图之间的比较要容易一些，顶点数不相同的 Delaunay 图之间的比较要复杂一些。顶点数相同的 Delaunay 图也便于图及图谱的方法的进一步处理，所以在每个极角方向选取唯一的点来构造 Delaunay 图，最终可以得到顶点数相同的 Delaunay 图，即大小相同的 Delaunay 图。

针对第一种情况，可以取所有相同方向的边缘点的平均位置作为该方向的顶点。

针对第二种情况，如果某个方向上没有边缘点，使用  $(\alpha + \Delta\theta)$  角度范围内的方向上所有边缘点的平均位置点作为该方向的顶点，如果仍然不能获得，就用  $(\alpha + 2\Delta\theta)$  角度范围内的方向上所有边缘点的平均位置点作为该方向的顶点。最坏的情况是在  $(\alpha + 2\Delta\theta)$  角度范围内的方向上都没有边缘点，这时候把形心作为该方向的顶点。 $\Delta\theta$  的值由经验给出。

为了解决可能存在的特殊情况，从整个物体形状中抽取特征点时采用的最小弧度间隔是  $\pi/32$ ，换句话说，能够从形状中自动提取的最多特征点数是 64 个，可以建立 64 个顶点的 Delaunay 图。根据以往的经验，并考虑到计算的复杂性和图的同构问题，30—100 个特征点用来构造的 Delaunay 图大小比较合适。

在极坐标系统下，根据极角来所选取轮廓的边缘点作为图的顶点，然后建立 Delaunay 图。当然，选择的间隔  $\Delta\theta$  越小，对形状的描述越精确。称这样建立的形状关联图为极图。当然如果不要求产生大小相同的图，例如仅仅使用图谱的主分量特征值来作为描述形状的特征向量。

当通过物体形状轮廓点构造出 Delaunay 图后，根据中介绍的邻接矩阵和邻接矩阵的图谱计算方法，就可以得到物体形状的极图谱描述。目标就是对图像中的形状建立对应的特征向量。根据图谱理论，图的谱特征主要包括主分量特征值、特征模体积、特征模周界、常数、模间邻接矩阵和模间距离等特征。通过分析各种图谱特征的有效性和计算复杂度，选取最简单且十分有效的主分量特征值（图的谱  $F_k = (|\lambda_k^1|, |\lambda_k^2|, \dots, |\lambda_k^n|)$ ）来建立代表每个形状极图的特征向量。

## 4.3 基于离散傅立叶变换的轮廓特征

### 4.3.1 傅立叶变换方法描述

草图的外围轮廓是封闭的，具有周期性，可以使用傅立叶变换进行描述其特征。首先将图像轮廓参数化得到复信号  $z = \langle z_0, z_1, \dots, z_{N-1} \rangle$  中，其中  $Z_k = x_k + jy_k (j = \sqrt{-1})$ ， $x_k$  和  $y_k$  分别是图像轮廓上的第  $k$  点的横、纵坐标。信号  $Z$  的离散傅立叶变换 (DFT) 如 4-18 式所示，其逆变换 (IDFT) 为如 4-19 式所示。其中  $R_n$ 、 $\theta_n$  分别表示 DFT 第  $n$  个系数的振幅和相位<sup>[46,47]</sup>。

$$Z_n = \sum_{k=0}^{N-1} Z_k e^{-j\frac{2\pi kn}{N}} = R_n e^{j\theta_n} \quad (n, k = -N/2, \dots, -1, 0, 1, \dots, N/2-1) \quad 4-18$$

$$Z_k = \frac{1}{N} \sum_{n=0}^{N-1} Z_n e^{j\frac{2\pi kn}{N}} \quad (n, k = -N/2, \dots, -1, 0, 1, \dots, N/2-1) \quad 4-19$$

### 4.3.2 不变性处理

物体形状在发生变化，如平移，旋转，缩放以及选择轮廓起始点不同，提取的特征向量要

保持不变，即要满足图形的平移，旋转，缩放以及起点选择无关性的处理。傅立叶变换本身的特性可以在频域对傅立叶变换的系数进行归一化，标准化过后的 DFT 系数  $Z_m = R_m e^{j\theta_m}$  满足各种不变性，以此作为物体轮廓的形状描述子，并且将其称为标准化形状描述子。

#### (1) 平移

对于平移，傅立叶变换后系数  $Z_0$  称为直流分量，包含了物体在图像中的位置信息，可以把  $Z_0$  归一化为 0 如 4-20 式所示。

$$Z_0 = 0 \quad 4-20$$

#### (2) 缩放

可以根据  $R_1$  做归一化处理如 4-21 式所示。

$$R_m = R_m / R_1 \quad 4-21$$

#### (3) 旋转和轮廓起点选择

假设物体的轮廓函数为  $z$ ，发生绕中心旋转  $\theta_0$  和起始点偏移  $l_0$ ，得到了新的  $z'_l = z_{l-l_0} e^{j\theta_0}$ ，做傅立叶变换：

$$Z'_m = Z_m e^{j\theta_0} e^{-j\frac{2\pi l_0 m}{N}}$$

于是得到  $\theta'_m = \theta_m + \theta_0 - \frac{2\pi l_0 m}{N}$ ，特别的当  $m$  为 1 和 -1 的时候， $\theta'_1 = \theta_1 + \theta_0 - \frac{2\pi l_0}{N}$  和

$$\theta'_{-1} = \theta_{-1} + \theta_0 + \frac{2\pi l_0}{N},$$

$$\begin{aligned} \bar{\theta}'_m &= \theta_m - \frac{\theta'_{-1} + \theta'_1}{2} + m \frac{\theta'_{-1} - \theta'_1}{2} \\ &= \theta_m + \theta_0 - \frac{\theta_{-1} + \theta_1}{2} - \frac{2\pi l_0 m}{N} - \theta_0 + m \frac{\theta_{-1} - \theta_1}{2} + \frac{2\pi l_0 m}{N} \\ &= \theta_m - \frac{\theta_{-1} + \theta_1}{2} + m \frac{\theta_{-1} - \theta_1}{2} = \bar{\theta}_m \end{aligned}$$

所以对旋转和轮廓起始点不同的归一化处理如 4-22 式所示。

$$\bar{\theta}_m = \theta_m - \frac{\theta_{-1} + \theta_1}{2} + m \frac{\theta_{-1} - \theta_1}{2} \quad 4-22$$

### 4.3.3 维数选择

将轮廓图像离散化为参数点，然后使用傅立叶变换，提取轮廓的模与相位特征。傅立叶变换提取轮廓的模与相位信息，可以通过傅立叶逆变换对轮廓进行重构。

由于边界轮廓点比较多，生成的特征向量维数会特别的大。同样大小不同的轮廓，生成特

征向量的维数如果不同，相似性比较会比较麻烦。由于物体的大致形状信息集中在低频部分，所以仅保留 DFT 变换后低频部分的  $M$  ( $M \ll N$ ) 个系数，即  $m=-M/2, \dots, -1, 0, 1, \dots, M/2-1$ 。这样大小不同的轮廓，生成特征向量的维数一样，都是  $M$  维特征向量。

$$E(X) = \sum_{M=-X/2, M \neq 0}^{M=X/2-1} |Z_M|^2 \quad 4-23$$

对于  $M$  的选择，选择越大，获得的系数就越多，傅立叶逆变换后得到的物体轮廓点也越多，同样进行相似性匹配的时间也越多。为了确定  $M$  的值，Bartolini L<sup>[46]</sup>通过系数所代表的能量大小来考察，然后选取一个经验数值。定义能量函数  $E(X)$  如 4-23 式所示。使用 SQUID 的 FISH Database 为例，计算  $E(M)/E(N)$ ，如图 4.6 所示。可以看到当选择  $M \in [16, 64]$  时，能量比变化范围在 84%—93%。当要选择  $E(M)/E(N) = 99\%$ ，需要  $M$  至少为 512，这个数值相对后面形状表示将会得到 512 轮廓点，将构造出 512 个顶点的图，显然不合适。所以选择一个相对较小的  $M$ ，比如  $M=32$ ，选择的  $M$  是 2 的幂，这样可以方便使用快速傅立叶变换。

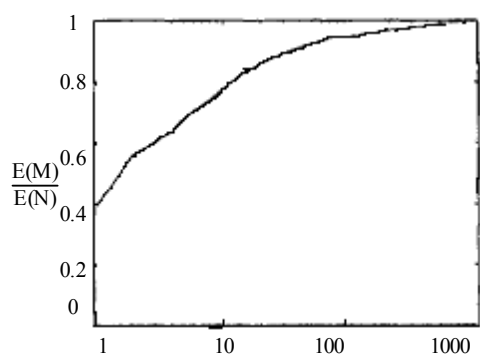


图 4.6 能量函数  $E(x)$

实验中，选取  $M=32$ 。并且傅立叶逆变换生成的降维后的图像为 32 个点。对于描述轮廓特征来说，在使用傅立叶变换时候时间开销不高，同时在增加  $M$  的值的时候，对于轮廓描述没有多少提高。傅立叶降维效果如图 4.7 所示，当  $M=32$ ，降维得到 32 个点的轮廓图。

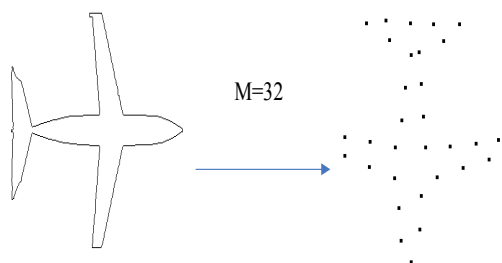


图 4.7 傅立叶降维效果



### 4.3.4 特征向量的生成

设质心为  $(x_0, y_0)$ ，那么质心距离  $r_i$  计算如 4-24 式所示。

$$r_i = \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2} \quad 4-24$$

对原来的轮廓进行傅立叶变换，选取低维系数，然后进行傅立叶逆变换重构降维后的轮廓。本文中  $M=32$ ，利用生成的 32 个外围轮廓点，生成 32 维的多边行。然后计算每个点到形心的距离，生成 32 维的特征向量。

### 4.3.5 实验结果分析

对狗的轮廓加上适当的变换，然后进行傅立叶变换，提取其特征，画出其对应的特征向量。图显示了狗原图 4.8，背部加粗图 4.9，尾巴加长图 4.10，尾巴剪裁图 4.11，傅立叶降维后的图以及对应的特征向量。从图中可以看出，对狗的原图进行背部加宽，尾巴剪除生成的特征向量图差别不大，而对于尾巴加长变换，生成的特征向量有很大的不同。原图上的特征向量图上特征点 A 与尾巴加长特征向量图上点 a 的相位相差较远，那是因为尾巴加长变换生成的离散点图相对与原图的点图进行了 180 度旋转。而实际上 A 点与 a 点对应后，B 点也与 b 点对应，两个图的特征向量也会非常相似。

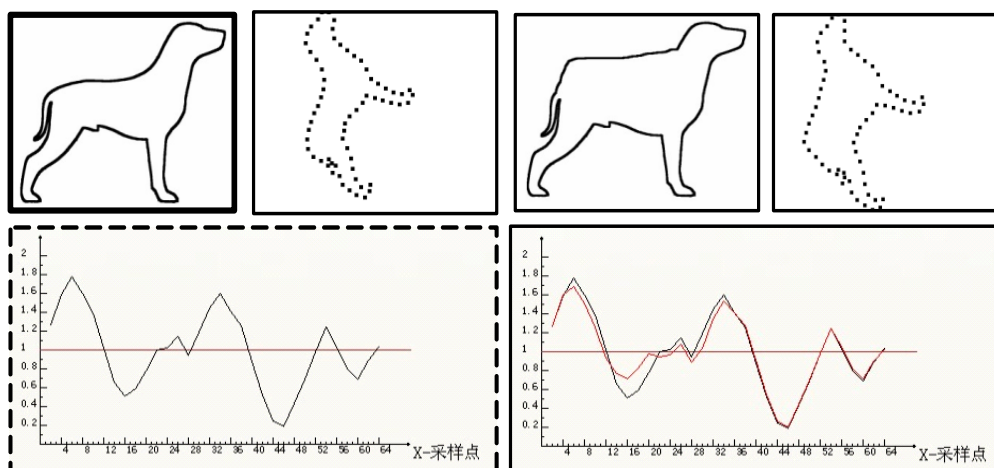


图 4.8 狗轮廓特征向量

图 4.9 背部加宽图和原图特征对比

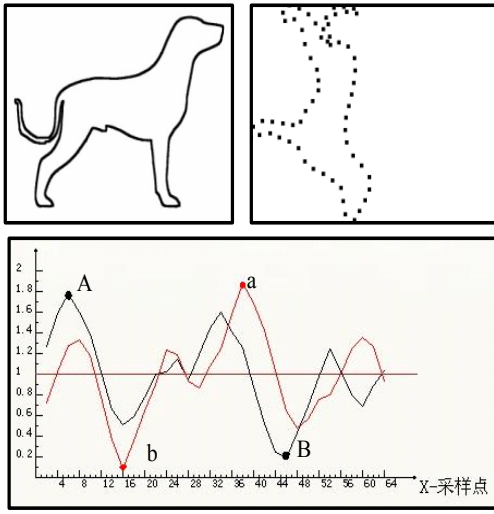


图 4.10 尾巴加长图和原图对比

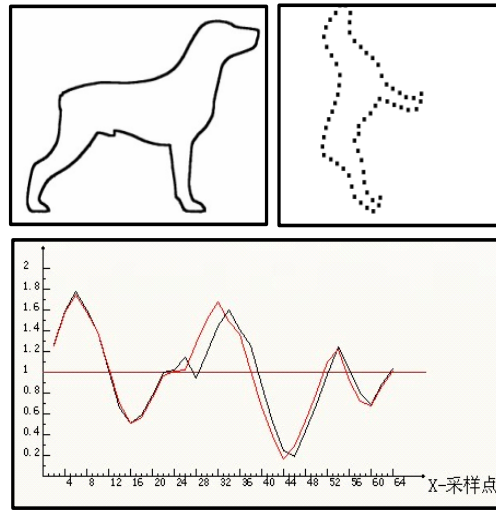


图 4.11 尾巴剪除图和原图对比

图 4.10 表明傅立叶变换可能对噪声具有一定的敏感性，对生成的特征向量做平移变换，生成一个新的特征向量解决噪声敏感的问题。比如对于特征向量  $R = (r_1, r_2, \dots, r_N)$ ，设  $r_k = \max(r_1, r_2, \dots, r_N)$ ，即  $r_k$  为最大一个维，那么新的特征向量为  $R' = (r_k, r_{k+1}, \dots, r_N, r_1, \dots, r_{k-1})$ 。那么相似性计算就需要同时计算  $R_A$  与  $R_A$  的距离和  $R_A$  和  $R_A$  的距离，选择其中小的作为特征向量之间的距离。在匹配效率增加的同时，也增加了计算特征向量之间距离的时间开销。在综合考虑效率和时间开销之后，本文采用保留原特征向量  $R$  以及以最大一维  $r_k$  归一化的特征向量  $R'$  作为轮廓的特征向量的方法，进行相似性计算。

## 4.4 基于多边形的轮廓特征

### 4.4.1 多边形近似

在一些应用中，数字图像边界提供的信息具有一定的冗余度，这时可以考虑用近似方法描述边界。最常见的方法是逐段线性近似方法(多边形近似方法)。这种方法用一个与原边界曲线相近的多边形来表示边界。为了获得满意的近似效果，可采用某种误差指标来衡量多边形与原曲线相近的近似程度。假设曲线  $L$  由一系列点  $A_1, A_2, \dots, A_N$  组成，连接起点  $A_1$  和末点  $A_N$  成直线  $A_1A_N$ 。曲线  $L$  可以用直线  $A_1A_N$  来近似，点  $A_i$  的坐标为  $(X_i, Y_i)$ 。设直线  $A_1A_N$  的方程为  $ax + by + c = 0$ ， $d_i$  为从  $A_i$  出发点到直线  $A_1A_N$  的距离，其定义如 4-25 式所示。

$$d_i = \frac{|aX_i + bY_i + c|}{\sqrt{a^2 + b^2}} \quad (i = 2, 3, \dots, N-1) \quad 4-25$$

直线  $A_1A_N$  近似曲线  $L$ ，有一定的误差，误差越小，表示的越精确。一条曲线  $L$  是否可以用直线近似，需要给定一定的拟合误差作为标准。近似度标准可以用均方误差  $E_2$ （如 4-26 式

所示)或者最大误差  $E_{\max}$  (如 4-27 式所示) 标示:

$$E_2 = \sum_{i=2}^{N-1} |d_i|^2 \quad 4-26$$

$$E_{\max} = \max |x_i - d_i| \quad 4-27$$

用多边形近似曲线时,也可以使用同样的近似标准当然也可以使用其它多边形逼近准则,如最大内部面积、最小外部面积、最小面积偏差等等。为了使多边形近似达到最优,在选择多边形顶点时应使总体误差最小。利用迭代搜索方法可以解决这个问题,但是计算复杂性高。

近似多边形的顶点数目在很大程度上取决于误差阈值。误差阈值越小,需的直线段就越多。多边形逼近主要有两种方法,分裂算法和合并算法。在分裂算法中,如果曲线是闭合的,起点  $A_1$  和终点  $A_N$  要选择曲线上相对两侧上的点,如图 4.12 这两个点将曲线分为两段。分裂算法的优点是可以找到曲线的转折点,并用这些转折点来描述曲线。而合并算法用相反的方式进行多边形近似,如图 4.13 所示。从曲线的一点  $A_1$  出发,沿顺时针方向或逆时针方向遍历曲线。对于每一点  $A_i$ ,检测  $A_1$  到  $A_i$  的误差,如果超过某一阈值,就将  $x_i$  作为多边形顶点并以  $x_i$  为起点,重复上述过程。如果曲线是闭合的,则选择曲线上外凸的转折点作为  $x_1$ 。

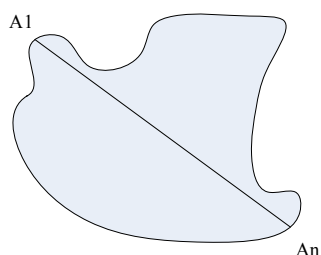


图 4.12 封闭边界的多边形近似分裂算法

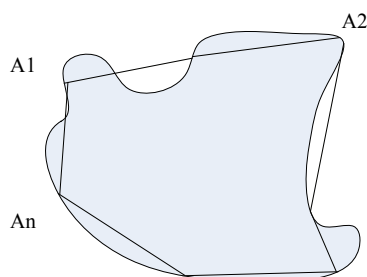


图 4.13 封闭边界的多边形近似合并算法

#### 4.4.2 多边形特征提取

**多边形顶点数** 多边形的顶点数越多,表示的形状越复杂。对于两个点数相差悬殊的多边形,有理由认为其形状相差很大。

**多边形内角平均值** 内角平均值是内角特征的一个总体描述,它能区分诸如三角形和正方形等形状。

**多边形内角的标准方差** 内角标准方差是对内角分布情况的一个总体描述,多边形越规则,其值越小。因此,可以区分规则多边形和不规则多边形等形状。

**多边形内角直方图** 所谓内角直方图,就是将  $(0, 2\pi)$  等角度或者不等角度划分为  $k$  个区间,然后统计落在这个区间上内角的数目。内角直方图是描述多边形内角分布的特性。

**紧凑度 LLS** 紧凑度定义如式 4-28 所示,表示轮廓图形和圆相似的程度。 $L$  表示多边形的

边长,  $S$  表示多边形的面积。LLS 的值越大, 说明形状越接近圆形。而且当 LLS 的值为 1 的时候, 形状为圆。

$$LLS = 4\pi S / L^2 \quad 4-28$$

## 4.5 本章小节

本章主要介绍草图特征提取过程, 草图特征提取方法。草图特征的提取要适合三维模型检索的需要, 可以快速的检索三维模型。提取轮廓的粗略特征, 可以将轮廓大致分类, 实现索引。轮廓的粗略特征包括紧凑性、轮廓上任意两点的距离和任意三点三角形的面积。轮廓的粗略特征可以用来快速排除非常不相似的轮廓。本文使用四种基于轮廓线的方法描述草图的轮廓特征: 等点采样、极图谱、离散傅立叶变换和多边形特征。

## 第五章 三维模型轮廓特征提取及相似性计算

三维模型特征提取有多种方法，但是提取的特征不能直接应用于基于手绘草图的三维模型检索。用户通过绘制草图检索需要的三维模型，因此必需提取三维模型的二维特征。把三维模型转化到二维空间，提取相应的特征向量，然后检索需要的三维模型。用户绘制的草图是三维模型在某个视点下的投影图，而用户选择的视点是不确定的，所以需要选择多个视点获取三维模型的投影轮廓。一般的三维模型特征提取之前要进行预处理，满足平移，旋转和比例缩放不变性。

相似性计算是通过计算特征向量之间的距离，查找用户需要的三维模型。三维模型投影轮廓的特征是包括多个方面的，投影轮廓的特征包括外围轮廓特征和内部基本图元特征。其中外围轮廓和手绘草图的外围轮廓一样，使用四种方法提取特特征。三维模型和手绘草图的相似性由多个特征之间的相似性加权和得到。

### 5.1 三维模型预处理技术

#### 5.1.1 预处理的必要性

对于三维模型检索系统，检索的准确性是系统的关键指标之一，要尽量提高三维模型的检索准确性。检索准确性不仅与三维模型几何相似性比较算法相关，还与三维模型检索系统中的一些预处理技术相关。通常使用的预处理技术包括三维模型坐标系归一化预处理和各向同性预处理。

三维模型坐标系归一化预处理用于统一所有三维模型的坐标系。对于三维模型检索系统来说，同样的三维模型在不同的坐标系下应该具有相同的相似度。若对坐标系尚不统一的模型进行特征提取并匹配，将对检索准确性有较大影响。因此在进行特征提取前必须对数据库中所有模型进行坐标系归一化预处理。三维模型的坐标系归一化预处理包括了三个方面，坐标系原点归一化预处理、坐标系比例归一化预处理、坐标系方向归一化预处理。

三维模型各向同性预处理也是提高三维模型检索系统的检索准确性的一项预处理。对三维模型进行特征计算之前进行各向同性预处理<sup>[48]</sup>，可以提高三维模型几何相似性比较算法的准确性。因此在特征提取前必须对三维模型进行各向同性预处理，把三维模型变为各向同性的三维模型。

#### 5.1.2 坐标系原点归一化

坐标系原点归一化问题即在每个模型上找一个确定的点作为新坐标系的原点，重新计算模

型在新坐标系下的坐标参数。一般认为重心平移的方法<sup>[49]</sup>对于坐标系原点归一化具有较好的鲁棒性。还可以采用面积加权的方法来提高重心平移的鲁棒性<sup>[50]</sup>，并证明这种方法可以减小三维模型顶点的采样密度，效果较好。具体计算如 5-1 式所示。

$$\begin{cases} \bar{P} = \sum_i P_i \cdot S_i \\ S_i = \sum_j S_j \end{cases} \quad 5-1$$

其中  $\bar{P}$  是三维模型重心， $S_i$  是每个相邻顶点的平均面积， $S_j$  为每个顶点片面的面积。

### 5.1.3 坐标系比例归一化

对于坐标系比例归一化问题，首先计算所有顶点的最大半径，然后对最大半径距离进行归一。这种坐标系比例归一化方法具有较好的鲁棒性。具体方法是，计算三维模型顶点和原点的最大距离  $R_{\max}$ 。把  $R_{\max}$  归一化，得到三维模型的缩放比例系数  $1/R_{\max}$ ，统一三维模型大小，统一后的三维模型可以用一个单位球包围。

### 5.1.4 坐标系方向归一化

坐标系方向归一化预处理是三维模型坐标系归一化最难解决的问题。李忠月对这方面技术进了综合分析和评价<sup>[51]</sup>。很多文献采用主元分析对三维模型进行坐标系方向归一化。主元分析是一种常用的信号分析方法，常用于信号的压缩领域，它通过对信号进行正交矩阵变换，将原始数据集合变换到主分量空间，使得单一数据样本的互相关性降低到最低点。

在三维模型检索中常用的方法有如下几种：

(1) 一阶矩方法。它首先根据三维模型点集的协方差矩阵计算出相应的特征值，按降序排列  $(\lambda_1, \lambda_2, \lambda_3)$ ，求出其对应的特征矢量  $(V1, V2, V3)$ 。V1 代表模型顶点分布最广的方向，即第一根主轴，V2 和 V3 分别表示模型顶点分布次要的第二根和第三根主轴，通过把特征向量进行转置后按行排列来构造变换矩阵，对三维模型进行坐标变换。变换后，模型的点集以  $(V1, V2, V3)$  为新的坐标系。这种方法计算简单，但对模型简化或细分的鲁棒性不好。

(2) 面积加权 PCA 方法。给模型的每个顶点，增加一个影响因子  $W_i$ ，( $W_i$  为  $V_i$  邻域面片的面积之和与模型所有面片的总面积之比)，此方法与上一方法相比减轻了模型简化或细分对模型归一化带来的影响。

(3) 改进 PCA 方法。一个模型完整的坐标归一化过程可以表示为如下的几何变换  $\tau(v) = s^{-1} \cdot F \cdot R \cdot (v - c)$ ，其中  $v$  表示给定模型的所有三角形的点集；把模型的质心移到原点  $c$  可以实现平移不变性； $R$  为 PCA 变换，在一个标准坐标平面中对齐模型，以确保旋转不变性； $F$  为翻转变换，以保证互为镜面映像的三维模型在表示方式上的同一性； $s^{-1}$  为比例变换，将旋转后的三维模型归一化到标准单元大小，以保证比例不变性。变换作用在初始模型的所有

顶点上。此方法相比于一般的 PCA 应用，它计算所有三角形的积分和，而不是求孤立顶点的和。这样就能够更加完备地用相同的权重去考虑模型中所有的点，缺点是积分计算的时间开销略高。

### 5.1.5 三维模型各向同性变换

Kazhdan 等在文献<sup>[48]</sup>中提出了对三维模型进行各向同性预处理的观点。在物理学中，各向异性表示物体在各个方向的性质不一样，例如纯净的水晶在各个方向的导热率不一致。各向同性表示物体在各个方向的性质一样，例如普通玻璃在各个方向的导热率一致。Kazhdan 把各向同性和各向异性的观点应用于三维模型几何相似性比较中，在计算三维模型特征之前，对三维模型进行各向同性预处理，使得三维模型的某个特征在各个方向上大小一致。

设  $\bar{P}$  和  $\bar{Q}$  是两个点序列（如 5-2 式所示），定义这两个点序列之间二次距离 Diff 如 5-3 式所示。其中  $\alpha$  为一个比例系数，HORN 证明，当  $\alpha$  按照公式 5-4 计算时，Diff 取值最小。

$$\bar{P} = \{P_1, P_2, \dots, P_n\}, \bar{Q} = \{Q_1, Q_2, \dots, Q_n\} \quad 5-2$$

$$Diff = \sum_{i=1}^n \|\alpha p_i - q_i / \alpha\| \quad 5-3$$

$$\alpha = \sqrt[4]{\sum \|q_i\|^2 / \sum \|p_i\|^2} \quad 5-4$$

$$P_i = \{p_x^i, p_y^i, p_z^i\}, Q_i = \{q_x^i, q_y^i, q_z^i\} \quad 5-5$$

$$\alpha = \sqrt[4]{\sum \|q_x^i\|^2 / \sum \|p_x^i\|^2} \quad 5-6$$

$$Diff_x = \sum_{i=1}^n [(\alpha p_x^i - \frac{q_x^i}{\alpha})^2 + (p_y^i - q_y^i)^2 + (p_z^i - q_z^i)^2] \quad 5-7$$

设两个三维模型顶点序列  $\bar{P}$  和  $\bar{Q}$  的坐标分别为  $P_i$  和  $Q_i$ ，定义如 5-5 式所示，两个模型在 X 轴方向的二次距离定义如 5-7 式所示。那么，可以推导出，当  $\alpha$  为公式 5-6 的时候，X 轴方向的二次距离  $Diff_x$  最小。当对这两个三维模型进行 X 轴方向的  $\alpha$  比例变换后，那么这两个三维模型在 X 轴方向二次距离为最小。这时，X 轴方向的比例系数  $\alpha$  为 1，对于 X 轴方向的  $\alpha = 1$ ，称这两个三维模型在 X 轴方向是同性。如果可以计算两个三维模型在各个方向的最优变换比例，使得三维模型顶点序列在各个方向的二次距离最小。当对三维模型进行各个方向的比例变换后，这两个三维模型在各个方向的  $\alpha = 1$ ，称这两个三维模型是各向同性模型，这个变换称为各向同性变换，相应的处理称为各向同性处理。

## 5.2 三维模型转化为二维深度图

三维模型信号规则化最简单的办法是将其转换为图像，本章研究了一种基于深度图像的特

征提取方法，提出了在特征中按图像重要性分配权重的策略。同时，在深度图像生成以及图像特征提取等方面对以往算法都做了有益的改进，为利用二维手绘草图接口进行三维模型检索的性能提高做了铺垫。

### 5.2.1 基本思路

相对于二维图形检索，三维模型检索无论是特征提取方法，还是检索性能，都不可与之同日而语。因此，一个很容易想到的办法是将三维问题转化为二维问题，利用比较成熟的图形检索技术来完成相对复杂的三维模型检索。其基本原理为：如果要检索想要的三维模型，那么输入三维模型的轮廓特征，一个某个方向的投影视图，如果三维模型的某个方向的投影图像和用户输入的相似，则可认为它是用户想要的三维模型。系统依据相似程度的大小把结果从大到小返回给用户。目前有多种基于视觉图像的特征提取方法，分别有各自的优缺点：三视图投影法比较简单直观，但是检索性能差强人意；二维切片法的关键在于找到正确的切割方向，缺点在于对角度偏差非常敏感；光场法检索性能较好，缺点是计算量太大，一次三维模型匹配需要很多次二维图像比较。有鉴如此，本文提出了一种基于深度图像的特征提取算法，旨在计算量可以接受的范围内，尽量提高三维模型的检索效率。

### 5.2.2 深度图像生成

将三维模型投影为平面图像有多种方法，其中代表性的有两种：三视图正交投影和正十二面体投影。本文在正交投影的基础上提出了一种基于 PCA 坐标面切分的投影方式。步骤为：首先对三维模型进行方位归一化，然后分别利用  $X-Y$ ， $Y-Z$ ， $Z-X$  三坐标面切分该模型，如图 5.1 所示，形成三对剖分模型。最后从坐标面向其外包围立方体相应平面作平行深度投影，得到六个方向上的投影图像，如图 5.2 所示。

相比三视图正交投影，采用切分投影一个重要的原因在于：对于非对称模型，可以在相反的方向上提供两个小同的投影，从而提供更丰富的描述细节。小过从另一个角度看，对于相对于某坐标面对称性很强的三维模型，其对应两投影图像则会非常相似，从而造成额外的存储空间消耗。一个极端情况是模型相对三个坐标面都对称，这样存储空间会比三视图正交投影冗余小。



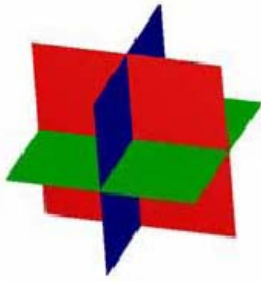


图 5.1 坐标平面的切分

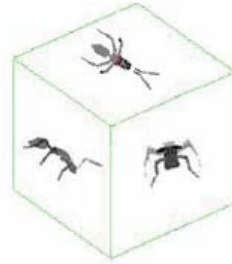


图 5.2 投影图像

首先计算三维模型的最小包围立方体，然后将其六个面  $\pi_1 \sim \pi_6$  都分为  $L \times L$  个像素单元，像素灰度级别为 0-255。投影的基本原则是：距离坐标而越远，像素灰度越小即颜色越浅。在对三维模型进行坐标面切分之后，针对包围立方体表面  $\pi_i$  的投影过程分为三个步骤：

- (1) 为  $\pi_i$  上的每个像素建立一个列表数据结构，用于记录可能覆盖它的三角面片。
- (2) 遍历对应剖分模型的所有三角面片，计算其在  $\pi_i$  上投影的最小包围盒，然后将它加入到包围盒内的所有像素的面片列表中去。
- (3) 对于  $\pi_i$  上的每个像素，遍历其面片列表，利用插值方式计算每一个面片在该像素位置距离面  $\pi_i$  的距离，选择最大值并将它转换为像素灰度值。

图 5.3 显示了三维猪模型坐标面切分对猪模型的投影图像，X+表示向 X 轴正方向投影，X-表示向 X 轴负方向投影，其余类推，第二行为切分后的模型，第三行为相应的投影图像。以此模型为例，除了 Z 轴方向的一组投影差别较小以外，X 轴和 Y 轴方向两组投影均有较大差别，因此本章方法比三视图投影包含更多的信息量。

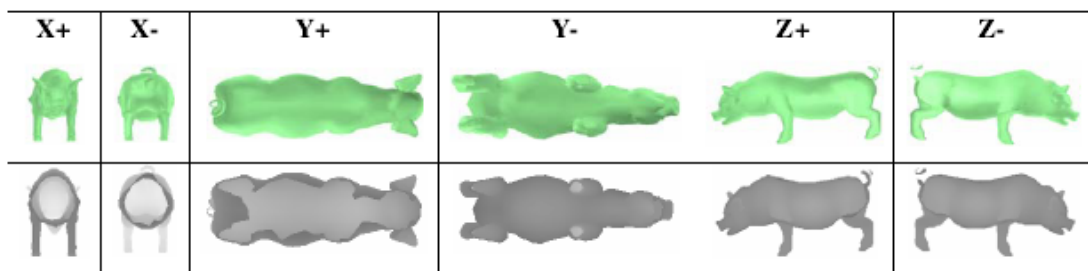


图 5.3 坐标面切分及投影图像

## 5.3 轮廓特征抽取

### 5.3.1 边缘提取

投影后得到的是投影图像，需要对投影的图像进行边缘提取，由于是三维轮廓的投影图像，所以轮廓提取方法和手绘的草图不一样，其实质是对图像进行边缘提取。边缘是指图像局部亮度变化最显著的部分。边缘主要存在于目标与目标、目标与背景之间，是图像分割、和形状特

征提取等图像分析的重要基础。图像中的边缘通常与图像亮度或图像亮度的一阶导数的不连续性有关。

图像亮度的不连续性可分为：（1）阶跃不连续，即图像亮度在不连续处的两边的像素灰度值有着显著的差异；（2）线条不连续，即图像亮度突然从一个值变化到另一个值，保持一个较小的行程后又返回到原来的值。对一个边缘来说，有可能同时具有阶跃和线条边缘特性。目的是找图像中灰度变化最剧烈的部分。经典的边缘提取方法是考察图像的每个像素在某个邻域内灰度的变化，利用边缘邻近一阶或二阶方向导数变化规律，用简易方法检测边缘。经典的利用一阶导数提取边缘的方法有 Roberts 算子、Sobel 算子等，利用二阶导数提取边缘的方法有拉普拉斯算子和 Marrs-Hildreth 算子等<sup>[52]</sup>。

#### （1）Roberts 算子

Roberts 交叉算子为梯度幅值计算提供了一种简单的近似方法，是一种利用局部差分算子寻找边缘的算子如 5-8 式所示。

$$G(i, j) = |f(i, j) - f(i+1, j+1)| + |f(i+1, j) - f(i, j+1)| \quad 5-8$$

用卷积模板标示方法，5-8 式变成 5-9 式。

$$G(i, j) = |G_x| + |G_y| \quad 5-9$$

用模板标示为 5-10 式。

$$G_x = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad G_y = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \quad 5-10$$

#### （2）Sobel 算子

Sobel 边缘算子由两个卷积核形成，图像中的每个点都用这两个核做卷积，一个核对通常的垂直边缘响应最大，而另一个对水平边缘响应最大。两个卷积的最大值作为该点的输出位。运算结果是一幅边缘幅度图像。其卷积模板表示如 5-11 式所示。

$$S_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 2 \end{bmatrix} \quad S_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad 5-11$$

#### （3）Prewitt 算子

Prewitt 算子（如 5-12 式所示）同 Sobel 算子一样，图像中的每一个点都用下面两个核进行卷积，取最大值作为输出。

$$S_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad S_y = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} \quad 5-12$$

#### (4) Kirsh 算子

Kirsh 算子由下面 8 个  $3 \times 3$  的窗口模板组成，每个小区域分别代表一种特定的检测方向。Kirsh 算子的卷积模板表示如下：

$$\begin{bmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix} \begin{bmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix} \begin{bmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{bmatrix} \begin{bmatrix} -3 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & 5 & -3 \end{bmatrix}$$

$$\begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{bmatrix} \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & 5 \\ -3 & 5 & 5 \end{bmatrix} \begin{bmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{bmatrix} \begin{bmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{bmatrix}$$

#### (5) 高斯—拉普拉斯算子

高斯—拉普拉斯算子是一种二阶边缘检测法，通过寻找图像的灰度值的二阶微分中的零穿越来检测边缘点，通常使用的拉普拉斯算子用模板卷积表示为：

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

#### (6) Robert 算子

Robert 算子是  $2 \times 2$  算子，对具有陡峭的低噪声图像响应最好。Sobel 算子、Prewitt 算子、Krisch 算子都是  $3 \times 3$  算子，对灰度渐变和噪声较多的图像处理得较好。Prewitt 算子并不是各向同性的，所以它检测出的边缘并不是完全连通的，有一定程度的断开；而使用 Roberts 和高斯—拉普拉斯算子就不存在这样的问题。通过对几种算法的分析比较，认为 Sobel 算子不仅实现容易、运算速度快，可提供最精确的边缘方向估计，且对噪声抑制能力较强。考虑工业分析系统的实时性要求，本方案采用 Sob 算子对边界进行检测。图 5.4 为边缘提取前后对比示意图。

### 5.3.2 轮廓跟踪

轮廓提取和轮廓跟踪的目的都是获得图像的外部轮廓特征。在必要的情况下应用一定的方法表达轮廓的特征，为图像的形状分析做准备。

二值图像轮廓提取的算法非常简单，就是掏空内部点：如果原图中有一点为黑，且它的 8 个相邻点都是黑色时（此时该点是内部点），则将该点删除。轮廓跟踪的基本方法是：先根据某些严格的“探测准则”找出目标物体轮廓上的像素，再根据这些像素的某些特征用一定的“跟踪准则”找出目标物体上的其它像素。下面来介绍两种二值图像轮廓跟踪的算法。

首先找到第一个边界像素的“探测准则”是“按照从左到右，从下到上的顺序搜索，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，按右、右上、上、左上、左、左下、下、右下的顺序找相邻点中的边界点 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 点继续找，直到找到 A 点为止。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。（即跟踪准则）。这种算法是要对每个边界像素周围的八个点进行判断，计算量比较大。下面来看另外一种跟踪准则。

首先按照上面所说的“探测准则”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。由于边界是连续的，所以每一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色，则为边界点，否则搜索方向顺时针旋转 45 度。这样一直到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<sup>[53]</sup>。

通过轮廓跟踪可以得到零件轮廓的链码，为下一步特征提取打下基础。轮廓跟踪实现了从无序边缘表示形成有序边缘表示的过程。图 5.5 为轮廓跟踪前后对比示意图。



图 5.4 边缘提取前后对比

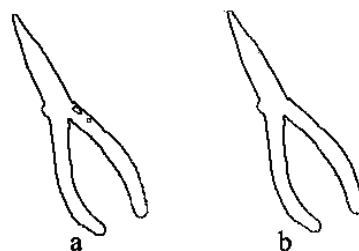


图 5.5 原始图像和轮廓跟踪图像比较

## 5.4 相似性计算

特征向量的相似性计算方法有多种，比如 2.5 节讲到的  $L_p$  距、 $\chi^2$  统计距和动态时间规整距  $d_{tw}$ 。不同方法抽取的特征量或者特征向量适合不同的距离匹配方法。 $L_p$  距是常用的方法，可以用来计算两个特征向量之间的距离，而且方法也比较简单。如果特征向量的每一维有不同方法生成的，而且需要平衡向量中每一维在相似性计算中的比重，可以用  $\chi^2$  统计距。为了适应轮廓有微小的变形，可以使用动态时间规整距  $d_{tw}$ 。本文使用多种特征向量提取方法描述手绘草图的特征，手绘草图的特征向量和三维模型的特征向量之间的距离用 DIFF 表示，是各种特征向量相似性计算的加权之和。

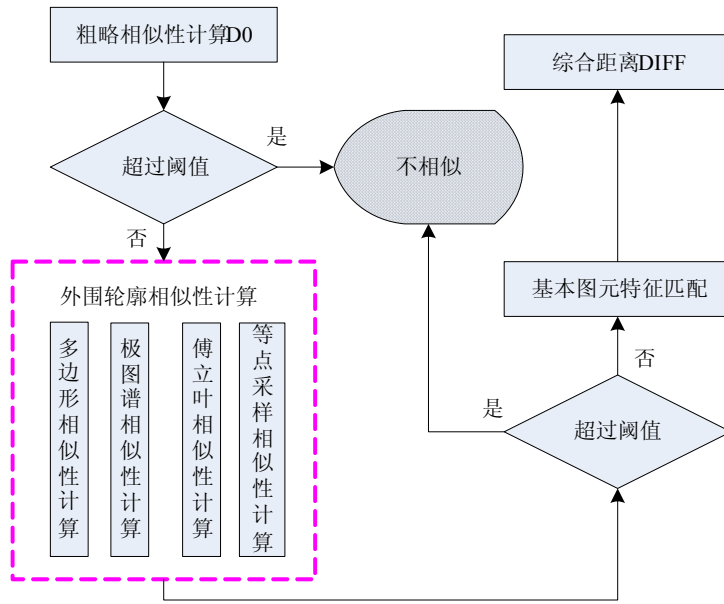


图 5.6 相似性计算流程

手绘草图特征用  $\bar{A}$  来表示，由三个特征向量组成，分别为  $T_1(\bar{A})$ 、 $T_2(\bar{A})$  和  $T_3(\bar{A})$ 。 $T_1(\bar{A})$  代表粗略的特征，包括任意两点间距离、任意三点之间的三角形面积和轮廓面积和周长平方的比值。 $T_2(\bar{A})$  代表外围轮廓特征，包括等点采样特征向量、傅立叶变换特征向量、极图谱特征向量和多边形特征向量。相应的模型的特征用  $\bar{B}$  来表示。那么手绘草图和三维模型之间的相似性计算如 5-13 式所示。

$$DIFF(\bar{A}, \bar{B}) = \sum_{i=1}^3 W_i L_i(T_i(\bar{A}), T_i(\bar{B})) \quad 5-13$$

$DIFF(\bar{A}, \bar{B})$  计算流程如图 5.6 所示。首先使用轮廓的粗略特征进行相似性计算，把距离相差很大的轮廓去掉。超过阈值代表轮廓差别很大，那么就不需要进行进一步相似性计算。然后匹配外围轮廓特征，看是否超过阈值，超过阈值代表不相似。没有超过阈值的进行基本图元信息比较。然后综合粗略特征、外围轮廓特征和基本图元特征，按照综合距离 DIFF 大小将结果排序返回给用户。

其中  $L_2(T_2(\bar{A}), T_2(\bar{B}))$  表示外围轮廓特征相似性计算，由等点采样特征、傅立叶变换特征、极图谱特征和多边形特征表示。 $L_2(T_2(\bar{A}), T_2(\bar{B}))$  计算如 5-14 式所示。

$$L_2(T_2(\bar{A}), T_2(\bar{B})) = \sum_{i=1}^4 W_{2,i} L_{2,i}(T_{2,i}(\bar{A}), T_{2,i}(\bar{B})) \quad 5-14$$

$$W_{2,i} = Precision_i(0.5) / \sum Precision_j(0.5) \quad 5-15$$

图 5.7 为多特征检索结果图，从图中可以看出，等点采样和傅立叶变换结果比较好，而极图谱和多边检索结果精度相对低一些。还可以看出，傅立叶变化比较快，当查全率比较高的时

候迅速下降。 $L_2(T_2(\bar{A}), T_2(\bar{B}))$  的相似性计算是由这个四个特征组合得到，那么可以通过检索结果图确定权值  $W_{2,i}$ ，如 5-15 式所示。其中  $Precision_i(0.5)$  表示第  $i$  种算法在查全率为 0.5 的时候检索的精度。

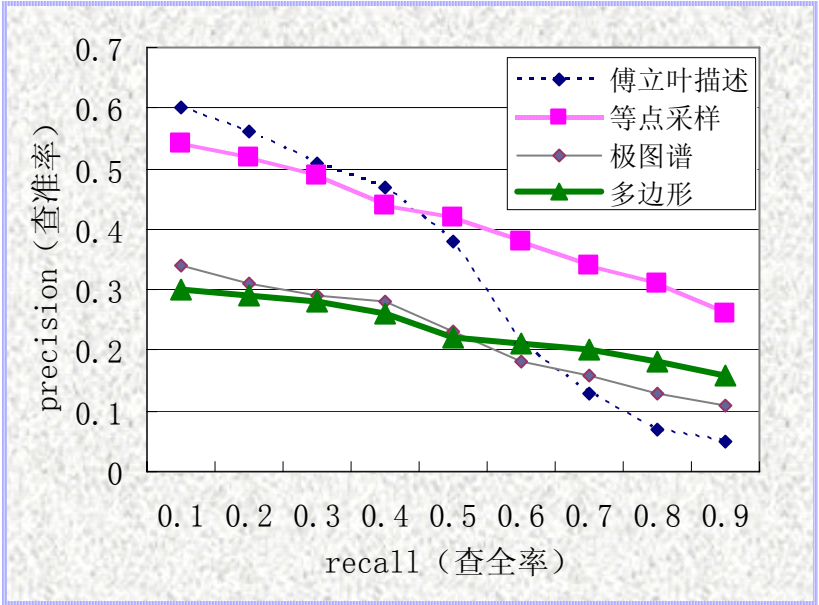


图 5.7 多特征检索结果比较

### 5.5 本章小节

本章主要介绍了三维模型轮廓特征提取方法和相似性计算问题。为了能有效检索三维模型，一般的系统模型库都要经过预处理。通常使用的预处理技术包括三维模型坐标系归一化预处理和各向同性预处理。坐标系归一化处理包括原点归一化处理、比例归一化处理和方向归一化处理。将处理后的三维模型，在坐标轴方向上投影，提取对应的投影轮廓。相似性计算是利用相应的特征向量比较草图和三维模型相似程度，特征向量的距离越小代表越相似。为了提高检索的效率，需用户对检索的结果进行评价，进一步了解用户的检索需求，重新检索三维模型。

## 第六章 基于手绘草图的三维模型检索系统实现

### 6.1 引言

前面章节对基于手绘草图的三维模型检索系统的框架及其关键技术进行了研究，本章利用前面介绍的算法实现基于手绘草图的三维模型检索系统。通过提取输入草图的外围轮廓特征及其拓扑特征，检索相关的三维模型，然后用户对系统检索的结果进行反馈，生成新的查询向量进行新的检索。基于手绘草图的三维模型检索系统是基于内容的三维模型检索系统的一部分，本文实现的基于手绘草图的三维模型检索系统属于基于内容的三维模型检索系统的二维接口。

就通用的三维模型检索方法来说，研究人员在世界各地研究出了许多的检索策略。如何对这些方法进行一个标准的评价，这是基于内容的三维模型检索领域的一个急需解决的问题。一个通用的标准测试平台会极大的方便该领域的合作、促进三维模型检索技术的发展和检索系统的应用。虽然现在国内、国外都有这方面的研究，但是遗憾的是，到现在还没有建成这样一个标准的测试平台。测试平台所应包含的内容一个标准的三维模型集、若干作为基准的查询样本、查询结果模型相对于查询样本的相关性评估值、检索策略性能评价方法。

本文使用一个简单的数据库作为测试数据库，实验数据所使用的三维模型，从因特网上下载得到。含有飞机、汽车、鱼、蛙等主要三维模型，共 224 个，同时包括少量的人头、花朵及狗等 78 个模型。实验中包含多种数据模型，适合作为本文的测试用例。

本系统使用的开发工具为 ASP.NET，选择 C# 语言；硬件为 Pentium 4 CPU 2.50 Ghz，DELL 微机；操作系统为 Windows xp2003；数据库为 SQL Server 2000。

### 6.2 系统功能结构和体系结构

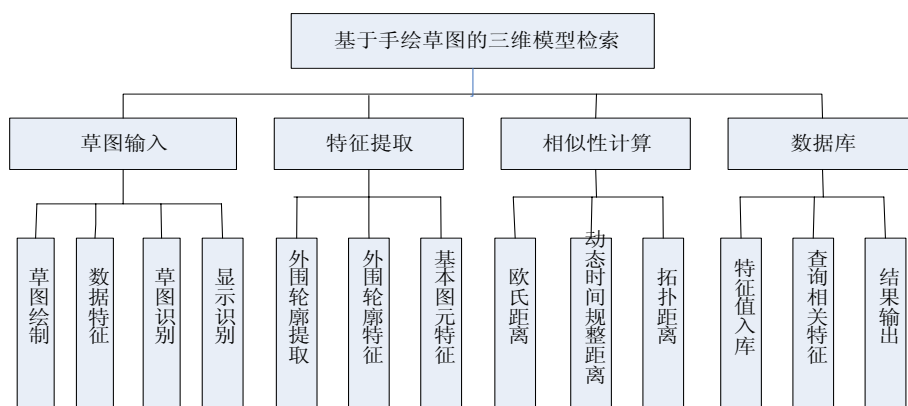


图 6.1 系统功能结构

在分析草图特征提取和相似性计算方算法后，制定并实现了系统功能如图 6.1 所示。

(1) 草图输入：本系统的数据输入有两种方式，一个是通过系统界面直接绘制草图，然后进行特征提取和相似性计算。另一个是通过读取硬盘上的草图文件，比如画画出来的图形或者其它画图软件画出来的草图，读取文件后提取草图的相关特征，然后进行相似性计算。

(2) 数据特征：数据特征是指绘制草图的时候，记录绘制点的一些速度，曲率等特征，利用相关的算法识别顶点，用于对用户输入的草图进行识别。

(3) 草图识别：草图识别就是利用用户绘制草图时，抽取草图上的特征点，然后识别用户的输入。包括噪声消除、去尾处理、直线拟合等。

(4) 显示识别：要把草图识别的结果反馈给用户，看是否已经正确识别了用户所输入的草图。对于识别反馈，用户可以提高探测点的数量，修改草图。通过和用户交互，更加准确的识别用户的输入意图。

(5) 外围轮廓提取：外围的轮廓特征是三维模型的重要特征，提取外围轮廓的特征生成特征向量用于三维模型检索。

(6) 外围轮廓特征：本系统使用三种方法描述外围轮廓：方法一是用傅立叶变换生成傅立叶描述子，生成特征向量；方法二是用多边形描述方法，等点生成固定维数的多边形，利用极心矩生成特征向量，然后对多边形进一步描述，利用最少的点近似描述轮廓，得到的多边形点数，内角平均值，多边形边长标准方差。方法三是使用关联图谱标示轮廓特征。

(7) 基本图元特征：基本图元有直线、弧线、曲线、圆、椭圆、多边形。先识别草图的外围轮廓，然后对其内部基本图元进行识别，提取基本图元的特征。

(8) 欧氏距离：对于固定维数的特征向量进行相似性匹配，一般可以使用欧氏距离进行相似性计算。

(9) 动态时间规整距离：动态时间规整距离（ $d_{tw}$  距离）匹配两个向量的距离，对于模型经过一定变换具有很好的检索效果。但是时间开销较高，时间复杂度为  $O(M^2)$ 。通常定义窗口的长度  $w$ 。计算两向量的  $d_{tw}$  距离的时候，两向量的维数之差不超过  $w$ ，那么时间复杂度降低为  $O(w \times w)$ 。

(10) 拓扑距离：利用识别的基本图元，计算对应基本图元之间的关系，进行拓扑距离匹配。

(11) 特征值入库：将输入的三维模型进行特征提取后，将其与已有的三维模型比较，从而实现三维模型快速分类。

(12) 查询相关特征：每个投影轮廓都有一定特征，利用索引查找数据库中可能相似的投影轮廓，然后进行相似性匹配，从而避免和数据库中每个投影轮廓都进行匹配的缺点。

(13) 结果输出：把符合用户查询需要的模型按照相关度从高到低排列显示给用户，同时用户可以对输出的结果进行评价，进行二次反馈。



按照系统功能要求，设计系统结构如图 6.2 所示。

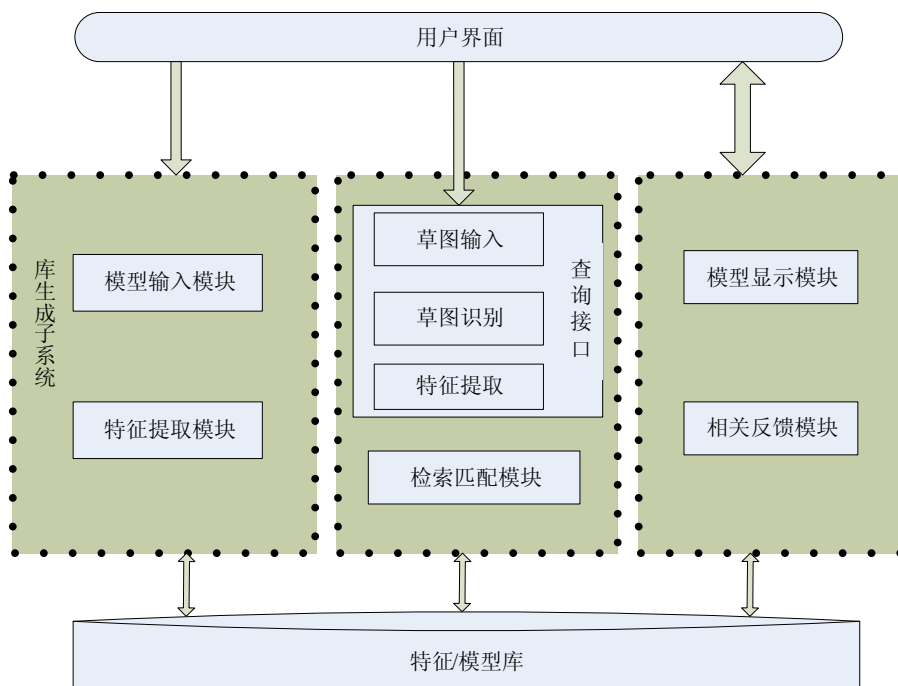


图 6.2 系统结构

设计的基于手绘草图的三维检索系统由三个子系统构成，库生成子系统、检索子系统和显示子系统组成。库生成子系统模块有模型输入模块和特征提取模块构成。三维模型输入是对入库前的模型进行管理及文本描述。通过模型输入模块，用户可以向系统输入三维模型，扩大三维模型库规模。对于输入的三维模型，除去文本描述以外，还需要对三维模型进行处理提取其特征，然后存入系统三维模型库。特征提取是库生成子系统的核心模块，它主要完成三维模型形状特征的提取。查询接口子系统完成基于形状的检索功能，由草图输入模块、草图识别模块、特征提取模块三个模块组成。用户通过查询接口可以对阈值、模型显示的数目查询方式等参数进行设定。检索匹配模块主要是完成当前用户输入的草图和系统三维模型的相似性匹配工作，得到模型之间相似度的大小并将检索到的模型的数据传送到显示子系统。显示模块主要用来显示检索的结果，包括模型显示模块和相关反馈模块。

### 6.3 数据库设计

检索系统数据库分为特征库和模型库。三维模型的特征值使用第三章介绍的方法提取并保存在数据库中的特征库的一个对应表单中，此表单还有关于对应三维模型的一些其它信息如文件名和三维模型文件路径等。在模型库中保存着描述三维模型的信息，如三维模型大小等。系统中数据库表设计如图 6.3 所示

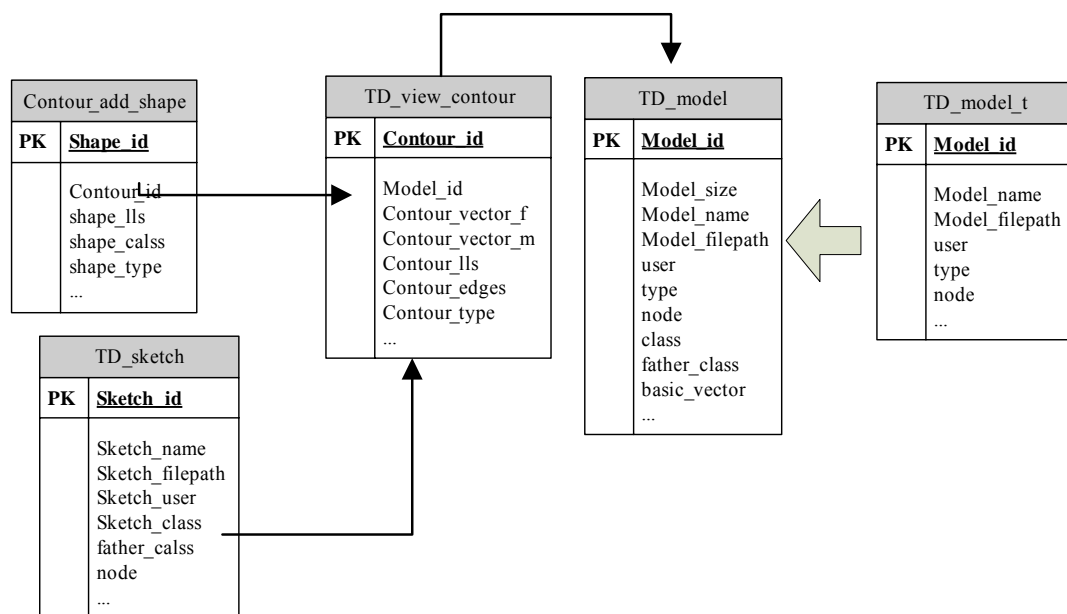


图 6.3 数据库设计

基于手绘草图的三维模型检索系统中，各表的基本信息简要介绍如下。

TD\_model 表是三维模型表，首先把系统模型存入数据库，把模型的一些基本信息存储在数据库的表里边。Model\_name 存放是三维模型名称，Model\_filepath 是存放的三维模型数据信息，三维模型的原始数据。User 代表上传模型的人，模型的来源。Type 是三维模型的存储类型，不同的三维模型存储的类型不一样，当前有多种软件可以用来生成三维模型。Class 代表三维模型的类别，比如按照某种分类把三维模型分成若干类别，这样有利于查找三维模型。基于手绘草图的三维模型检索系统一般属于基于内容的三维模型检索的二维接口部分，三维模型的数据表应当还包含其它一些字段用于三维模型检索系统的扩充，但是本系统只用到其中部分字段。

TD\_model\_t 表和 TD\_model 表一样用来存储三维模型，但是存放的是一些临时的三维模型，当系统比较空闲的时候可以调用三维模型处理模块，提取三维模型的特征，存入数据库，把模型信息从 TD\_model\_t 转换到 TD\_model 中。TD\_model 实际上是存储没有经过处理的三维模型。

Td\_view\_contour 表是三维模型投影轮廓特征表，Model\_id 是外键，记录是哪个三维模型的投影轮廓。Contour\_id 是主键，Contout\_vectour\_f 是由傅立叶描述子生成的特征向量，Contout\_vectour\_m 是有固定多边形生成的特征向量。Contout\_edges 是最少边近似轮廓后生成的多边形，记录其边数。Contour\_lls 是轮廓面积的  $4\pi$  倍和边长平方的比值。Contour\_type 是轮廓的形状大致分类，如果是已知常见的类型，那么检索会比较容易处理。Contour\_type 描述的是其凹凸特性。

TD\_sketch 表是存储用户检索时绘制的轮廓，这样可以记录一个用户的绘制图形的习惯，按照该用户的历史习惯预测当前的绘制要求。同时可以把类似的草图调入过来供用户进行参考。

Contour\_add\_shape 表是存储用户手绘轮廓的内部图元基本信息，这些基本图元是属于某一外围轮廓补充信息。记录这个图元的大小，所属于的轮廓，轮廓面积的  $4\pi$  倍和边长平方的比值，大致的形状，如圆，三角形，四边形，线段，一般多边形等等。

## 6.4 关键技术实现

本系统的实现有三个关键部分，分别是适合于三维模型检索的草图识别、草图特征的提取和三维模型特征库的生成。由于系统实现的代码比较长，本文仅给出轮廓抽取和傅立叶变换的实现函数。

### 6.4.1 轮廓抽取代码示例

读取轮廓信息到数组 b\_map，数组 b\_map 定义为 `bool[,] b_map = new bool[h + 2, w + 2]`。对数组 b\_map 处理提取其轮廓放入数组 b\_map\_ctrl 中，数组 b\_map\_ctrl 定义为 `bool[,] b_map_ctrl = new bool[h + 2, w + 2]`。按照点提取的顺序，把点存入 b\_position 数组中，数组定义为 `int[,] b_position = new int[10000, 2]`。

轮廓信息获取过程如下：

```
int[,] drct = new int[16, 2] { { 0, 1 }, { -1, 1 }, { -1, 0 }, { -1, -1 }, { 0, -1 }, { 1, -1 },
    { 1, 0 }, { 1, 1 }, { 0, 1 }, { -1, 1 }, { -1, 0 }, { -1, -1 }, { 0, -1 }, { 1, -1 },
    { 1, 0 }, { 1, 1 } };

for (i = 0; i < h + 2; i++)
    for (j = 0; j < w + 2; j++)
    {
        //找到第一个点，那么就返回，
        if (b_map[i, j] == true)
        {
            x_point = i;
            y_point = j;
            break;
        }
    }

if (x_point == 0 && y_point == 0)
{
    return false;
}
else
{

```

```

//把第一个点加入轮廓线点集合

b_map_ctrl[x_point, y_point] = true;

b_position[num, 0] = x_point;

b_position[num, 1] = y_point;

num++;

}

flag = true;

int direction = 4, ls_direction;

int n = 0;

while (flag)

{
    //获取外围轮廓 //x_pint,y_point代表第一个点，d代表一个方向
    for (n = 0; n < 7; n++)
    {
        ls_direction = direction + n;

        i = x_point + drct[ls_direction, 0];

        j = y_point + drct[ls_direction, 1];

        if (b_map[i, j] == true)
        {
            direction = (ls_direction + 5) % 8;

            x_point = i;

            y_point = j;

            break;
        }
    }

    if (b_map_ctrl[x_point, y_point] == true)
    {
        flag = false;

        break;
    }

    else
    {

```

```

        b_map_ctrl[x_point, y_point] = true;

        b_position[num, 0] = x_point;

        b_position[num, 1] = y_point;

        num++;

    }

} //while(flag)

```

#### 6.4.2 傅立叶变换代码示例

进行傅立叶变换，生成特征向量，定义数组 GF，DFT，IDFT 分别为傅立叶变换的初始点坐标、傅立叶变换后坐标，傅立叶逆变换后坐标。其中 GF 由 b\_position 数组转化得到，b\_position 为 int，而 GF 为 float 类型。代码如下：

```

double jiao;    //代表角度的。

double xk, yk, xx, yy; //代表实部及其虚部。

    for (n = 0; n < ds; n++)

    {

        xk = yk = 0;

        for (k = 0; k < NUM; k++)

        {

            jiao = Math.PI * (-1.0) * (n - ds / 2) * (double)k * 2 / (double)NUM;

            xx = GF[k, 0];

            yy = GF[k, 1];

            xk += xx * Math.Cos(jiao) - yy * Math.Sin(jiao);

            yk += xx * Math.Sin(jiao) + yy * Math.Cos(jiao);

        }

        DFT[n, 0] = xk;

        DFT[n, 1] = yk;

    }

    for (n = 0; n < ds; n++)

    {

        IDFT[n, 0] = Math.Sqrt(DFT[n, 0] * DFT[n, 0] + DFT[n, 1] * DFT[n, 1]);

        IDFT[n, 1] = Math.Atan2(DFT[n, 1], DFT[n, 0]);

    }

```

```

xk = IDFT[ds / 2 - 1, 1] + IDFT[ds / 2 + 1, 1];
yk = IDFT[ds / 2 - 1, 1] - IDFT[ds / 2 + 1, 1];
for (n = 0; n < ds; n++)
{
    IDFT[n, 1] = IDFT[n, 1] - 0.5 * xk + (n - ds / 2) * 0.5 * yk;
    IDFT[n, 0] /= IDFT[0, 0];
}
for (n = 0; n < ds; n++)
{
    DFT[n, 0] = IDFT[n, 0] * Math.Cos(IDFT[n, 1]);
    DFT[n, 1] = IDFT[n, 0] * Math.Sin(IDFT[n, 1]);
}
for (n = 0; n < ds; n++)
{
    xk = yk = 0;
    for (k = 0; k < ds; k++)
    {
        jiao = Math.PI * (n - ds / 2) * (double)k * 2 / (double)ds;
        xx = DFT[k, 0];
        yy = DFT[k, 1];
        xk += xx * Math.Cos(jiao) - yy * Math.Sin(jiao);
        yk += xx * Math.Sin(jiao) + yy * Math.Cos(jiao);
    }
    IDFT[n, 0] = xk / ds;
    IDFT[n, 1] = yk / ds;
}

```

对于 IDFT，生成特征向量，然后绘制特征向量图。 $\text{arrFourier}[i] = (\text{IDFT}[i * 2, 0] - xk) * (\text{IDFT}[i * 2, 0] - xk) + (\text{IDFT}[i * 2, 1] - yk) * (\text{IDFT}[i * 2, 1] - yk)$ 。这样特征向量就是保存在  $\text{arrFourier}[i]$  里边，可利用  $\text{arrFourier}[i]$  绘制特征向量图。

## 6.5 系统运行示例

用户可以通过鼠标绘制自己需要的草图，同样也可以调用系统中已经存在的草图文件。通

过系统绘制草图，系统可以和用户进行交互，把识别结果反馈给用户。用户根据反馈的结果，选择继续绘制还是修改草图。系统记录当前用户绘制草图的速度和曲率等信息。当然，用户也可以通过其它方法绘制草图，比如通过手绘笔绘制自己需要的草图，以文件的形式保存在系统草图文件夹中。用户还可以通过其它工具和软件绘制草图，比如通过画图工具绘制草图，然后保存在系统硬盘上。通过其它方式绘制的草图，不能获取其速度信息，只能通过曲率，采样点密度来提取草图中的特征点。

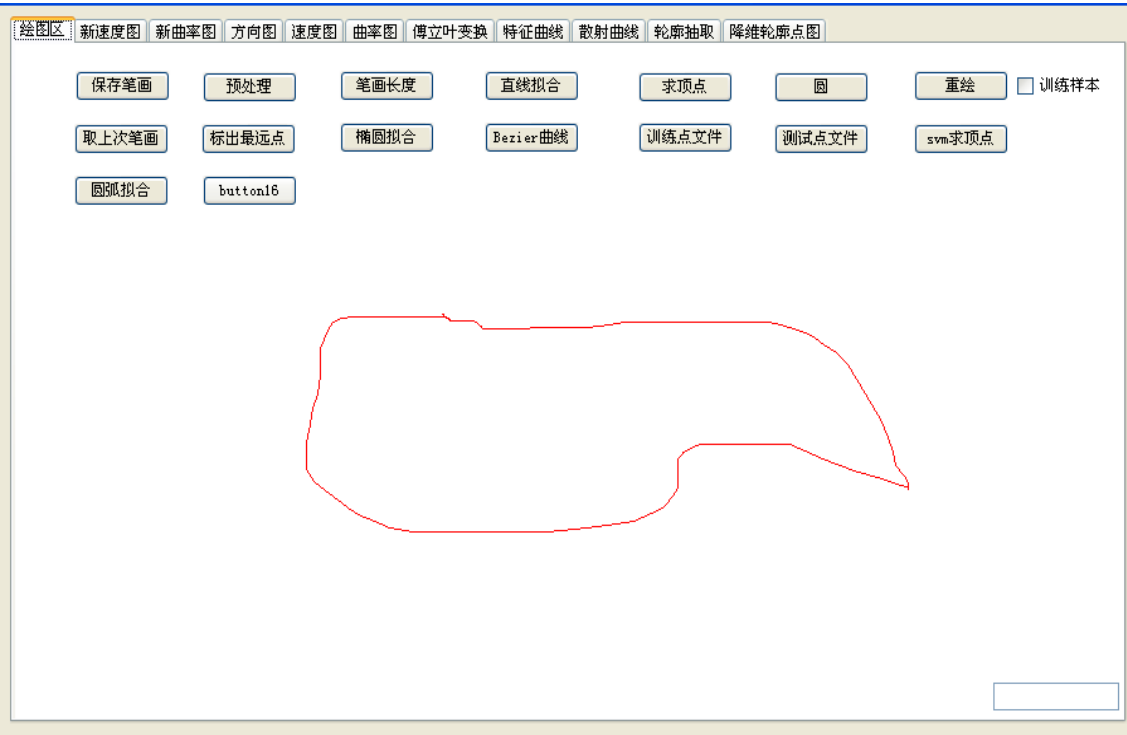


图 6.4 用户绘图界面

图 6.4 所示显示了在系统上自由绘制草图的界面。用户可以输入草图，然后点击预处理按钮，对草图进行预处理。然后点击保存笔画，把此笔画保存起来。如果用户想重新绘制自己的草图，可以点击重绘按钮。这样可以取消当前的绘制草图，然后重新开始绘制草图。

用户绘制的草图往往不是一次性完成的，用户可以确认当前输入的笔划，然后继续绘制草图。这样用户在绘制草图出错的时候，可以不用重绘，而是返回到刚才保存的笔划处继续绘制自己需要的草图。用户如果想保存自己当前的笔划，那么可以点击保存笔划按钮。

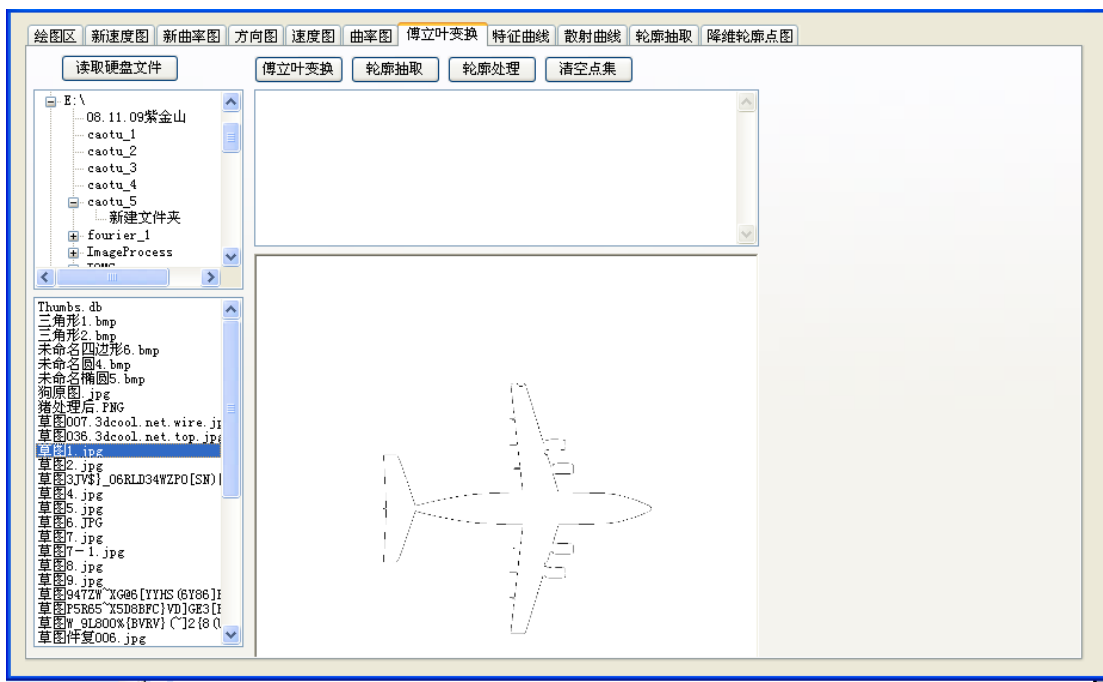


图 6.5 读入草图

图 6.5 显示用户从系统硬盘读取草图文件的界面。用户可以通过选择文件路径，选择需要的草图文件，对草图进行预处理。对于给定的轮廓，首先获取其轮廓，然后进行傅立叶变换。生成特征向量曲线图。

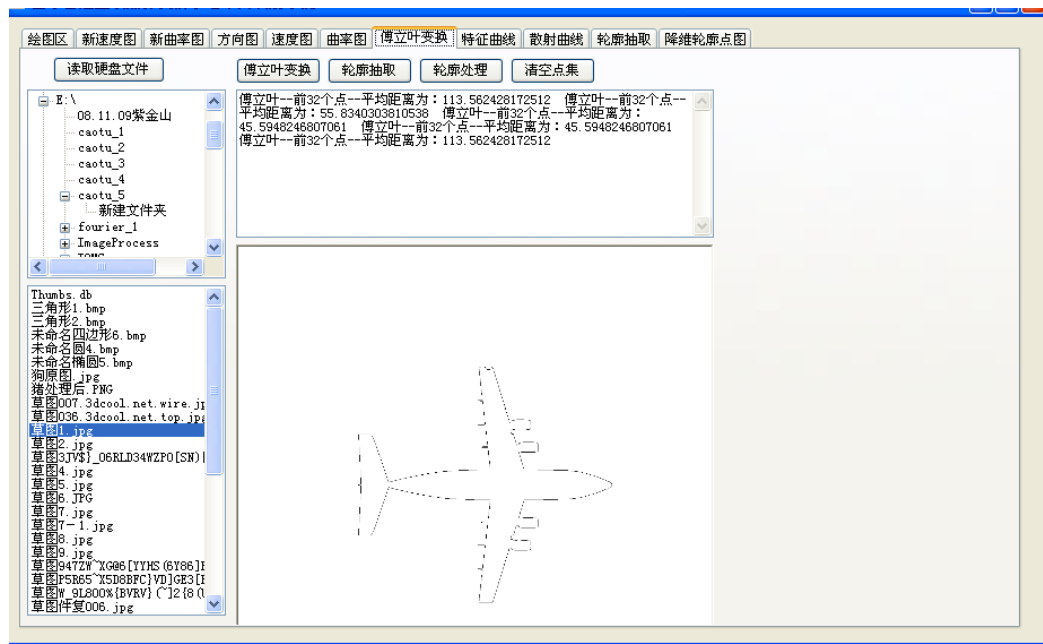


图 6.6 傅立叶变换飞机一

可以对比不同的轮廓生成的特征曲线，在同一坐标轴上显示，这样可以很清楚看出不同轮廓生成的差异情况。就是每次傅立叶变换后生成的特征向量都是保存在 `arrFourier[i]` 里边，



但是特征向量的元素  $i$  的存储位置为  $ds \times num + i$ 。  $ds$  为降维后点数，就是特征向量的维数，而  $num$  是当前已经存在轮廓的数量。

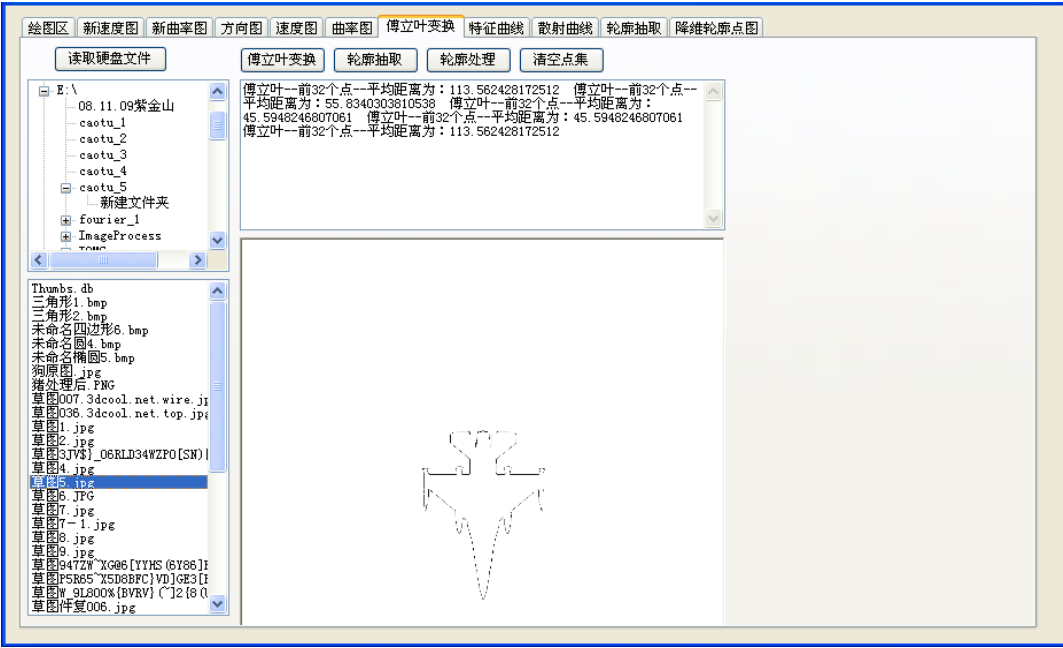


图 6.7 傅立叶变换飞机二

图 6.6 和图 6.7 为两个飞机轮廓，图 6.8 显示这两个飞机轮廓在傅立叶变换后生成的特征向量在同一坐标下的对比图。可以看出这两个飞机的特征向量比较相似，两个飞机方向也一致，说明傅立叶变换具有旋转不变性，生成的特征向量可以用于相似性比较。

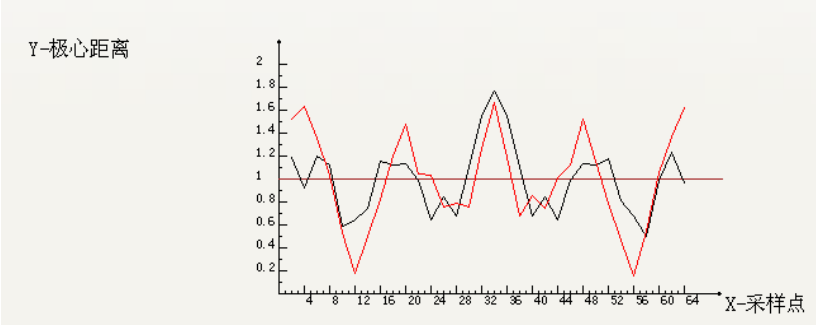


图 6.8 飞机一和飞机二特征向量比较

图 6.9 所示为用户通过鼠标在界面上手绘的飞机，然后通过草图预处理、特征提取和相似性计算检索所需要的三维模型。

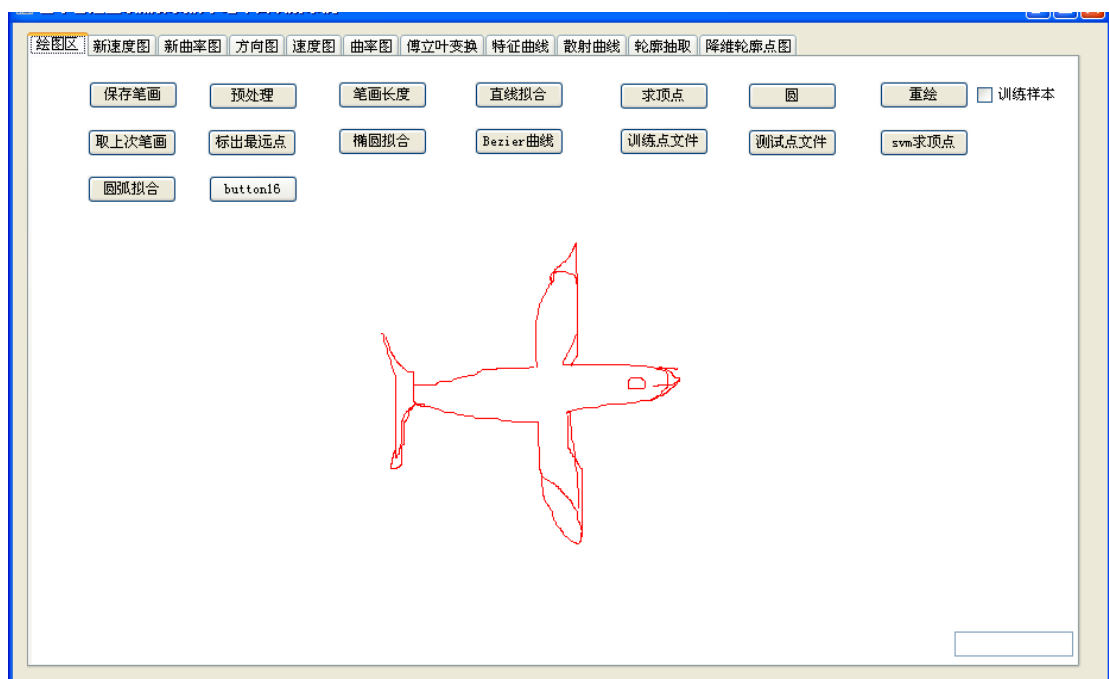


图 6.9 检索飞机输入示例

系统检索结果如图 6.10 所示。从检索结果看，可以检索相关的模型，同样也会检索到轮廓特征相似但是本身并不相关的模型。还可以检索到有旋转，大小不一的模型。可见基于手绘草图轮廓特征的 3D 模型检索系统中，可以满足 3D 模型的旋转，平移及缩放的不变性。



图 6.10 检索结果显示

## 6.6 本章小节

本章利用前面研究的算法，使用 C#.NET 实现系统。首先介绍系统的开发环境和使用的数据库，然后介绍系统的功能结构图，接着按照功能结构图实现系统结构图。系统使用的数据库是 SQL Server 2000，介绍了系统所用到的关键数据表。最后介绍系统各个模块运行情况，以及关键部分的代码。

## 第七章 总结展望

### 7.1 本文总结

基于内容的三维模型检索是近些年兴起的一个新的研究领域，本文研究的基于手绘草图的三维模型检索属于基于内容的三维模型检索的二维接口部分。本文提出基于手绘草图检索三维模型的一些方法，研究一些算法，把三维模型检索和图形检索的一些方法应用于基于手绘草图的三维模型检索。本文做的工作是：

(1) 草图识别。具体介绍了草图识别的基本原理，草图识别步骤和草图识别方法。首先对草图预处理，消除草图输入时候的噪声；然后进行相似性计算，提取手绘草图的特征点；最后把系统识别的结果反馈给用户。用户可以确认输入，或者重新输入，还可以在已经绘制的草图上继续绘制以增加更详细的需要检索的模型信息。草图识别的结果是草图的外围轮廓和内部基本图元信息。

(2) 草图特征提取技术。通过草图识别后，要对识别的基本图元进行特征提取。由于外围轮廓包含大量的信息，要对外围轮廓进行精确详细的描述。同时为了快速检索和索引的需要，需要粗略的提取草图的特征。草图的粗略特征包括紧凑性、任意两点间距离和任意三点的三角形的面积。本文使用四种方法描述草图的轮廓特征，分别是多边形方法、等点采样方法、极图谱方法和离散傅立叶变换方法。

(3) 三维特征库生成。由于三维模型库中模型数量比较大，每次检索三维模型都要进行特征提取需要大量的时间同时也不现实。可以通过在三维模型入库的时候，提取模型的特征存入系统的特征向量库。通过选择视点，对三维模型投影，把三维模型转化为二维表示的方法，提取其特征存入系统特征库。

(4) 相似性计算。相似性计算是计算两个特征向量之间的距离，距离越小代表越相近。由于草图的特征不是单一表示的，如何选择每个特征的权重是需要解决的问题。本文使用每个单一特征检索模型，然后考察在查全率为 50% 的时候查准率。某一特征的权重为此特征的查准率与其它所有特征查准率之和的比值。

最后，根据以上的研究实现了一个基于手绘草图的三维模型检索原形系统，本系统是具有良好扩展性的系统，实现了高速、有效的三维模型检索。

### 7.2 进一步研究工作

本文研究基于手绘草图的三维模型检索系统，只是使用了一些基本的算法和思想。本文对基于手绘草图的三维模型检索系统中的草图识别方法进行了初步的探索，如何使用更加有效的

方法识别草图需要进一步提高。基于内容的三维模型检索在理论方法和实现技术上都取得了一定的研究成果，如何把这些成果应用于基于手绘草图的三维模型检索是一个重要的问题。今后的工作，还将围绕以下几个方向进行：

(1) 草图识别模块应满足基于手绘草图的三维模型检索系统的需要。本系统中使用的草图识别和一般的草图识别系统不一样，本系统中的草图是用来描述三维模型投影轮廓的，不会有特别复杂的基本图形存在。应当在借鉴一般草图识别系统基础上，简化识别方法加快识别的速度实现符合本系统需要的草图识别模块。

(2) 有效的描述基本图元特征和进行相似性计算。外围轮廓包含大量的信息，同时内部基本图元也包含重要的信息。对于狗、牛、猪等动物模型。内部有意义的基本图元包括眼睛、鼻子、耳朵等投影后的基本图元。基本图元间的位置比较简单，相对独立，同时基本图元的数量也相对比较少。而对于建筑等复杂的三维模型，投影后基本图元信息丰富，角和特殊曲线的特征比较明显。如何通过基本图元信息把三维模型分类，从而快速的检索三维模型。同时，基本图元间的相对位置和相对关系比较简单，需要寻找更加适合的算法匹配基本图元特征。

(3) 寻找更加适合的投影方法。三维模型在不同的视点下得到不同的投影轮廓，而有些方向上的投影轮廓包含的信息少，不适合用来检索三维模型。表现在两个方面，一是用户在输入此模型轮廓的时候，所绘制的轮廓和此轮廓差别总是很大。二是用户在绘制轮廓进行检索的时候，和此轮廓特征比较接近，但是对应的模型往往不是用户需要的三维模型。如何选择更好的方法投影轮廓，包含在多个方向上的投影同时去除多余的投影轮廓。

(4) 轮廓形状描述。本文使用的是基于轮廓边缘的形状描述方法，描述轮廓形状的方法还有基于区域的方法。应当结合基于区域描述和基于轮廓边缘描述的优点结合使用，提高识别的精度。

(5) 建立有效的索引。建立有效的索引才可以快速的检索需要的三维模型。本文使用紧凑性把轮廓大致分类的方法比较粗糙。应当结合多种特征建立多维索引，检索需要的三维模型的时候只在一个比较小的三维模型空间搜索。

## 参考文献

- [1] Min P., Halderman A., Kazhdan M., Funkhouser A.. Early experiences with a 3d model search engine. In: Proceedings of Web3D Symposium, Saint Malo, France, 2003, 7-18.
- [2] Yamada A., Pickering M., Jeannin S., Cieplinski L., Ohm JR., Kim M.. MPEG-7 visual part of experimentation model version 9.0. MPEG Video Group, Pisa, Italy: Technical Report ISO/MPEG N3914, 2001.
- [3] 赵斌, MPEG-7 与多媒体信息描述, 图书馆杂志, 2005, 24(2): 48-52.
- [4] 单锦来, 陈博, 杨献春, 许满武, MPEG-7 和 MPEG-7 实验模型参考软件, 计算机科学, 2003, 30(6): 31-37.
- [5] Revankar S., Yegnanarayana B., Machine Recognition and Correction of Freehand Geometric Line Sketches. In: Proc. Of IEEE Intl.Conf. on Systems .Man, and Cybernetics.1991.1:87-92.
- [6] Xu X G, Sun Z X, Liu W Y. Matching Spatial Relation Graphs Using a Constrained Partial Permutation Strategy. Journal of Southeast University.2003, 19(3):236-239.
- [7] Takayuki D K, Ajay A. Van V. Recognizing Multi-stroke Geometric Shapes: An Experimental Evaluation. In: Proc. of 6th Annual ACM Symposium on User Interface Software and Technology, 1993.121-128.
- [8] Fonseca M J, Jorge J A. Using Fuzzy Logic to Recognize Geometric Shapes Interactively. In: Proc. 9th IEEE Conf. on Fuzzy Systems ,2000, 1:291-296.
- [9] 孙建勇, 手绘草图识别中的用户适应性研究, 南京大学申请硕士学位论文, 2004.5.
- [10] 孙正兴, 彭彬彬, 丛兰兰, 孙建勇, 张斌, 在线草图识别中的用户适应性研究, 计算机辅助设计与图形学学报, Vol.16(9), 2004, p 1207-1215.
- [11] 孙正兴, 徐晓刚, 孙建勇, 金翔宇, 一种面向方案设计的图形输入工具, 计算机辅助设计与图形学学报, Vol.15(9) , 2003, p 1145—1152.
- [12] 庄越挺, 潘云鹤, 吴飞, 网上多媒体信息分析与检索, 北京:清华大学出版社, 2002.
- [13] P. Min, A. Halderman, M. Kazhdan, and T. Funkhouser. Early experiences with a 3d model search engine. In: Proceedings of Web3D Symposium, Saint Malo, France, 2003, 7-18.

- [14] Tangelder, J.W., Velthkamp, R.C.: A survey of content based 3D shape retrieval methods. In: Proc. Shape Modeling International, Genoa, Italy, 2004:145-156.
- [15] Y. Liu, J. Pu, G. Xin, H. Zha, W. Liu, and Y. Uehara, A Robust Method for Content-based 3D Model Retrieval. Proc. Pacific Graphics, 2004, p 3-9.
- [16] Funkhouser T. Min P. Kazhdan M, et al. A search engine for 3d models. ACM Transactions on Graphics, 2003,22(1):83-105.
- [17] Funkhouser T., Min P., Kazhdan M. et al.. A search engine for 3d models. ACM Transactions on Graphics, 2003, 22(1): 83-105.
- [18] Zhang C., Chen T.. Efficient feature extraction for 2d/3d objects in mesh representation. In: Proceedings of IEEE International Conference on Image Processing, Thessaloniki, Greece, 2001,935-938.
- [19] Neal J., Shapiro G. A symbolic representation for 3d object feature detection. In: Proceedings of International Conference on Pattern Recognition, Barcelona, Spain, 2000, 1221-1224.
- [20] Cyr C., Kimia B.. 3d object recognition using shape similarity-based aspect graph. In: Proceedings of IEEE International Conference on Computer Vision, Vancouver, Canada, 2001, 254-261.
- [21] Tomas H. Vaclav S. A Survey of Methods for 3D Model Feature Extraction. Accepted in 5th Seminar Geometry and graphics in teaching contemporary engineer, Szczyrk, Poland, 2003.
- [22] Vranic, D. V., Saupe, D., Description of 3D-shape using a complex function on the sphere. In: IEEE International Conference on Multimedia and Expo (ICME 2002), Lausanne, Switzerland, 2002, pp. 177-180.
- [23] Brennecke A, Isenberg T. 3D Shape Matching Using Skeleton Graphs. Simulation and Visualisierung (SimVis), SCS European Publishing House, Erlangen, San Diego, 2004 299-310.
- [24] Min P., Haklerman A., Kazhdan M., Funkhouser A..Early Experiences with a 3d model search engine. In: Proceedings of Web3D Symposium, Saint Malo, France, 2003, 7-18.
- [25] D.J. Berndt and J. Clifford, "Using Dynamic Time Warping to Find Patterns in Time Series", Advances in Knowledge Discovery in Databases: Papers from the 1994 AAAI Workshop, July 1994, 359-370.

- [26] J. Rocchio, Relevance Feedback in Information Retrieval: The SMART Retrieval System, Prentice Hall Inc, New Jersey,USA,1971:313-323.
- [27] Salton G, McGill M J, Introduction to Modern Information Retrieval, New York: McGrawHill, 1983:8-9.
- [28] Rocchio Jr J J. Relevance Feedback in Information Retrieval. In The SMART Retrieval System: Experiments in Automatic Document Processing. Gerard Salton, Prentice-Hall, 1971,313-323.
- [29] Ishikawa Y, Subramanya R, Faloutsos C. MindReader: Query databases through multiple examples. Proc. of the 24th VLDB Conf.. New York, 1998, 218-227.
- [30] Rui Y, Huang T S. Optimizing learning in image retrieval. Proc. of the IEEE Int'l Conf. on Computer Vision and Pattern Recognition. Hilton Head Island, 2000, 236-243.
- [31] Lee C, Ma W Y, Zhang H J. Information Embedding Based on user's relevance Feedback for Image Retrieval. SPIE Photonic East. Boston, USA.1999.p 20-22
- [32] 朱兴全, 张宏江, 刘文印等.iFind:一个结合语义和视觉特征的图像相关反馈检索系统.计算机学报.2002, 25 (7):681-689.
- [33] M.L.Kersten, M.F.N.de Boer, Query Optimization Strategies for Browsing Sessions, Workshop on Foundations of Models and Languages for Data and Objects, Volkse, Germany, Springer- Verlag, 1993:69-84.
- [34] Bernard J. Jansen, Amanda Spink, Tefko Saracevic, Real life, real users, and real needs: a study and analysis of user queries on the web, Information Processing and Management: an International Journal, Amsterdam, Netherlands, Elsevier Science, 2000, 36(2):207-227.
- [35] John C.Shafer, Rakesh Agrawal, Continuous Querying in Database-Centric Web Application, WWW9 conference, Amsterdam, Netherlands, Computer Networks,2000, 33(1-6):519-531.
- [36] Sklansky J and Gonzalez V, Fast Polygonal Approximation of Digitized Curves, Pattern Recognition 1980,12(5):327-331.
- [37] Jin Xiangyu,Sketch-based user interface study, [A Thesis for the Application of Master Degree], Nanjing, China, Nanjing University,June, 2002.
- [38] Paulson, B., Hammond, T. Accurate Primitive Sketch Recognition and Beautification. In Proceedings of Intelligent User Interfaces (IUI'08), 2008.



- [39] Sezgin M T. Feature point detection and curve approximation for early processing of free-hand sketches[D]. Massachusetts: Massachusetts Institute of Technology, 2001, P 53-60.
- [40] Yu, B. and Cai, S. A Domain-Independent System for Sketch Recognition. In Proc. of the 1st International Conference on Computer Graphics and Interactive Techniques in Australasia and South East Asia, ACM Press (2003),141-146.
- [41] Delaunay 三角剖分算法, 百度百科, <http://baike.baidu.com/view/1691145.htm>.
- [42] A.J. Schwenk. Almost all trees are cospectral. New directions in the theory of graphs, New York: Academic Press, 1973. 275-307.
- [43] D.M. Cvetkovi'c, M. Doob, H. Sachs. Spectra of graphs-Theory and application, 2nd ed.[M]. Berlin: VEB Deutscher Verlag d.Wiss., 1982, P 153-179.
- [44] 范益政.图的谱理论[D]. Ph.D. thesis, 中国科学技术大学, 2001 年 10 月
- [45] E. Hückel.Quantentheoretische Beiträge zum Benzolproblem[J].ZPhys.2001, 70:204-286.
- [46] Bartolini L,Ciaccia P,Patella M.WARP:Accurate Retrieval of Shapes Using Phase of Fourier Descriptors and Time Warping Distance[J].IEEE Trans.Pattern Analysis and Machine Intel-ligence,2005,27(1):142-147.
- [47] Zhang Chunyan,Tang Jin,Luo Bin.Shape Edit Distance on Contour based Shapes[C] //IEEE Sixth International Confer-ence on Intelligent Systems Design and Applications. [s.l.]:[s.n.], 2006:310-315.
- [48] Kazhdan M., Funkhouser T., Rusinkiewicz S.. "Shape Matching and Anisotropy".ACM Transactions on Graphics (2004 SIGGRAPH), 2004, 23(3): 623-629.
- [49] Kazhdan M., Funkhouser T., Rusinkiewicz S.. "Rotation invariant spherical harmonic representation of 3D shape descriptors". ACM SIGGRAPH symposium on Geometry, June 2003, Session 6, 156-164.
- [50] Vranic D.V, Saupe D., Richter J.. "Tools For 3D Object Retrieval: Karhunen-Loeve Transform and Spherical harmonics". In Proceedings of IEEE 2001 Workshop Multimedia Signal Processings, 2001, 293-298.
- [51] 李忠月. “三维模型检索中的预处理技术”.计算机工程与设计,2006.5,1886-1888.

- [52] 崔屹, 数字图像处理技术及应用, 北京: 电子工业出版社, 1997, 66-72.
- [53] 何斌, 马大宇, 王运坚, 等, Visual C++数字图像处理, 北京: 人民邮电出版社, 2001, 435-444.

# 致谢

本文是在导师谢强副教授的精心指导下完成的。谢强老师治学严谨、学识渊博、思想深邃，他那宽以待人的崇高风范，朴实无华、平易近人的人格魅力，与无微不至的人文关怀，令人如沐春风，倍感温馨。在此我谨对谢老师几年来的辛勤培养和悉心关怀表示衷心的感谢，并祝愿导师合家欢乐，一生平安！

感谢丁秋林老师在三年里为我们做的精彩演讲，让我的思想得到了升华，明白了很多待人接物与为人处世的道理。

感谢郑洪源老师、周良老师在我几年的学习中给予的真诚帮助和悉心指导。

感谢教研室所有关心和帮助过我的同学，你们让我感受到了团结的力量并找到了前进的方向。感谢他们三年来给予的关怀和支持。他们的陪伴，使我度过了一个充实和快乐的研究生生涯。

感谢李雪峰和叶敬茂同学和我的讨论，让我的思路更加清晰，让我更有信心研究本课题。

感谢师妹张颖颖同学，抽出宝贵的时间帮我通读论文，给我提出宝贵的意见。

感谢张晓伟、马志国、刘宁和窦丹丹同学，三年中陪我一起进餐。饭前饭后的开心交流，共同探讨人生的美好与今后的目标。

感谢王勇、夏金月、陶运信同学，和你们一起度过我的研究生生活十分开心。共同的生活习惯让我的休息得到了充分的保证，相同的追求让我们在研究生阶段一起努力。

感谢我的父母、哥哥、姐姐和小妹。虽然家境不好，但是父母支持我完成学业。感谢哥哥、姐姐和小妹对我的经济支持。特别感谢我的小妹，感谢她多年来的全力支持！

## 在学期间的研究成果及发表的学术论文

[1] 张必欢, 谢强, 周良, 三维模型检索中手绘草图轮廓特征的应用研究, 中国制造业信息化, 2009. 1