

哈尔滨理工大学

毕 业 设 计

题 目 : 基于手绘草图的三维模型检索

院、系 : 计算机科学技术学院 计算机系

姓 名 : 邱泳锋

指导教师 : 高雪瑶

系主任 : 孙冬璞

2020 年 6 月 X 日

哈尔滨理工大学毕业设计（论文）评语

学生姓名：邱泳锋	学号：1704010213
学 院： 计算机科学与技术	专业： 计算机科学与技术
任务起止时间： 年 月 日至 年 月 日	
毕业设计（论文）题目：	
指导教师对毕业设计（论文）的评语：	
指导教师签名：_____ 指导教师职称：	
评阅教师对毕业设计（论文）的评语：	
评阅教师签名：_____ 评阅教师职称：	
答辩委员会对毕业设计（论文）的评语：	
答辩委员会评定，该生毕业设计（论文）成绩为：	
答辩委员会主席签名：_____ 职称：	
年 月 日	

哈尔滨理工大学毕业设计（论文）任务书

学生姓名：邱泳锋		学号：1704010213	
学 院：计算机科学与技术		专业：计算机科学与技术	
任务起止时间：2021 年 3 月 1 日至 2021 年 6 月 20 日			
毕业设计（论文）题目：基于手绘草图的三维模型检索			
<p>毕业设计工作内容：</p> <p>本系统采用 VS2019 + MFC + OpenGL + OpenCV + ImGui，以 C++ 为主要语言进行开发，计划实现模型检索，根据用户绘制的草图，检索出三维模型。</p> <p>进度安排：</p> <p>(1)2021.3.1-2021.4.15：开发最优视图集提取模块、全局视图特征提取模块和二维形状特征提取模块。</p> <p>(2)2021.4.16-2021.5.1：建立三维检索模型，开发相似性计算模块。</p> <p>(3)2021.5.2-2021.6.1：开发三维模型检索模块，计算草图与三维模型之间的相似性。撰写论文。</p> <p>(4)2021.6.2-2021.6.20：修改完善论文，准备答辩。</p>			
<p>资料：</p> <p>[1] Yuan Juefei, Abdul Rashid Hameed, Li Bo et al. A comparison of methods for 3D scene shape retrieval[J]. Computer Vision and Image Understanding, 2020, 201.</p> <p>[2] Zhu Z, Wang X, Bai S, et al. Deep learning representation using autoencoder for 3D shape retrieval[J]. Neurocomputing, 2016, 204: 41-50.</p> <p>[3]基于深度学习的三维形状特征提取方法[J]. 周燕,曾凡智,吴臣,罗粤,刘紫琴. 计算机科学. 2019(09):47-58</p> <p>[4]张云峰. 图像与三维模型匹配方法的研究及应用[D]. 南京：南京大学, 2019.</p> <p>[5] 安勃卿. 基于手绘草图的三维模型检索研究与实现[D].西北大学, 2017.</p>			
指导教师意见：		系主任意见：	
签名：		签名：	
年 月 日		年 月 日	

基于手绘草图的三维模型检索

摘 要

随着计算机 CPU 和 GPU 的计算能力大大提高, 三维模型不仅变得越来越复杂, 而且细节信息也越来越丰富, 在许多不同的领域应用得愈来愈广泛, 如动画相关领域、机械行业领域、医疗事业领域等, 三维模型的数量也越来越多, 而三维模型的分类和检索已成为一个重要的研究方向。尽管许多学者对三维模型检索技术进行了各种研究, 并提出了许多不同的模型检索算法, 但对于三维模型检索而言, 仍有许多问题有待解决。

本文通过对现有三维模型检索技术的研究分析, 发现三维模型固有的复杂性和高维度数据的计算会严重影响了模型的检索速度和准确性。本文采用如下方法来进行处理。首先, 对三维模型进行降维处理, 接着, 通过将二维视图作为三维模型的检索条件, 以降低检索成本。最后, 提出了一种基于手绘草图的三维模型检索方法。本文对以下的内容做了主要的研究。

为了提高检索的准确性, 首先通过适当的空间位置变化来渲染三维模型, 然后根据固定投影的方法得到二维视图图集。每个模型选择 6 个 2D 视图作为 3D 模型的特征视图集。然后从手绘草图和二维视图图集中提取特征向量, 并以 Zernike 矩和 Fourier 描述符作为集成特征描述符, 构造加权的全局视图特征和 D2 描述符集。通过集成特征描述符来检索三维模型, 以充当用户手绘草图和模型的二维视图的特征向量, 从而执行相似性评估。实验结果表明, 该方法能够有效地对三维模型进行分类。

关键词 三维模型检索; Zernike 矩; Fourier 描述符; D2 形状描述符; 集成特征描述符

3D model retrieval based on hand drawn sketch

Abstract

With the rapid development of CPU and GPU, 3D models are not only becoming more and more complex and rich in details, but also widely used in animation, machinery, medical and other fields. The number of 3D models is also increasing. The classification and retrieval of 3D models has become an important research direction. Although many scholars have done different researches on 3D model retrieval technology and proposed many different model retrieval algorithms, there are still many problems to be solved.

Through the research and analysis of the existing 3D model retrieval technology, this paper finds that the inherent complexity of 3D model and high-dimensional calculation seriously affect model retrieval. In this paper, the three-dimensional model dimension reduction method is used, and the two-dimensional view is used as the retrieval condition of the three-dimensional model to reduce the retrieval cost. This paper presents a 3D model retrieval method based on hand-drawn sketches.

In order to improve the accuracy of retrieval, firstly, the 3D model is rendered according to the appropriate spatial position change, and then the 2D view set is obtained according to the fixed projection method. Each model selects 6 2D views as the feature view set of 3D model. Secondly, feature vectors are extracted from sketch and 2D view sets, and weighted sets of global view features and D2 descriptors are constructed by Zernike moments and Fourier descriptors as integrated feature descriptors. By integrating feature descriptors to serve as feature vectors of user's hand drawn sketches and 2D views of the model, similarity evaluation is performed to retrieve 3D models. Experimental results show that this method can effectively classify 3D models.

Keywords 3D model retrieval, Zernike, Fourier descriptor, D2 shape descriptor, Integrated feature descriptor

目 录

摘要	I
Abstract	II
第 1 章 绪论	1
1.1 课题背景	1
1.2 研究意义	1
1.3 国内外研究现状	2
1.4 课题的主要研究内容	3
1.5 本文文章结构安排	3
第 2 章 基于手绘草图的三维模型检索的总体框架	5
2.1 三维模型检索的体系架构	5
2.2 三维模型检索的处理流程	6
2.2.1 三维模型投影的处理架构	6
2.2.2 三维模型检索的处理架构	7
2.3 本章小结	8
第 3 章 三维模型的处理	9
3.1 OFF 模型文件格式解析及渲染	9
3.2 固定视角的模型投影技术	10
3.3 三维模型的光照和材质添加	14
3.4 三维模型的空间变化	16
3.5 三维模型投影系统结果	17
3.6 本章小结	18
第 4 章 草图的绘制处理	19
4.1 直线类图形绘制	19
4.1.1 直线	20
4.1.2 矩形	20
4.1.3 三角形	21
4.2 曲线图形绘制	21
4.3 圆形绘制	22
4.4 铅笔线绘制	23
4.5 手绘草图的优化	23
4.6 草图绘制的结果	25
4.7 本章小结	25
第 5 章 草图及三维模型的特征提取	26

5.1 全局试图特征描述符	26
5.1.1 Zernike 矩	26
5.1.2 Fourier 描述符	28
5.2 D2 描述符	29
5.3 集成描述符	30
5.4 本章小结	32
第 6 章 基于手绘草图的三维模型检索	33
6.1 相似性计算	33
6.1.1 欧几里得距离	34
6.1.2 曼哈顿距离	34
6.1.3 切比雪夫距离	34
6.1.4 闵可夫斯基距离	34
6.1.5 马氏距离	35
6.2 基于手绘草图的三维模型检索结果	36
6.3 本章小结	37
结论	38
致谢	39
参考文献	40
附录 A	41
附录 B	48

第1章 绪论

1.1 课题背景

随着三维建模技术的进步，特别是三维扫描仪相关设备的逐步成熟和相对普及，当今的互联网上积累了大量的三维模型，这使得模型数量在迅速增加。举例来说，在模型交易平台 Turbosquid 中，用户可以直获取海量的不同的三维模型。在 Google 3D 仓库中上，用户可以从拥有超过上万个的模型中任意选择自己所需要的模型。为了合理且有效率的利用现有的三维模型，有必要开发一个便捷可靠的三维检索引擎。一个高效可靠的三维模型检索引擎是通过特定的交互操作从数据库中找到一个满足用户意图的三维模型，待检索的对象可以是整个模型（如整台电脑）或模型的一部分（如电脑的显示屏）。

目前三维模型在种类与个数上的增长的速率日渐增加，而且应用越来越广泛。三维模型广泛应用于较多的不同领域，如加工制造业产品的开发设计、建筑模型的电脑绘制、各类游戏的虚拟现实技术、计算机辅助开发和绘图、三维游戏的开发和影视动画特效等。不难发现，三维立体数据信息实验模型数据库转变得越来越普遍常见。在服务网络上，每一天都存在多种多样的三维立体实验模型形成。截至当前，MPEG-7 参考标准实验模型的一个分析研究主要内容，就是三维立体实验模型。综上所述，三维立体实验模型的搜索是现如今计算机图形检索的一个热点研究课题。

1.2 研究意义

相关三维建模企业应该优先把握在当前激烈的市场竞争中的商机，生产出符合用户需求的产品，并从中获取利润，就不得不思考降低三维模型设计和三维模型处理的高昂成本。研究和调查表明，在产品的设计过程中，只需从头开始制作约 20% 个设计。40% 设计可以通过直接重用现有设计来完成；此外，可以通过对现有设计进行适当的修改来获得 40% 设计。如果您在每次生产新产品时从头开始选择材料，绘制图纸并设计加工计划，这不仅会花费时间，而且会增加成本。在生产过程中，经常发生重复设计，其本质原因是没有高效的 CAD 模型检索工具。快速有效的 CAD 模型检索无疑将大大缩短开发周期，不仅提高了设计效率，而且为企业带来更多的效益。重用现有的 CAD 模型也将使企业节省更多的成本。

如今，市面上常见的 CAD 模型库的规模因受到三维 CAD 模型的种类

和数量的扩大而快速加大。据 Wikipedia 的不完全统计，目前世界上现存的三维模型将近有 300 亿个。随之而来的问题便是：三维模型的结构变得更加复杂多样。同时，国内外专家和工程师开始对三维 CAD 模型的检索研究投入了更多的精力，因为从模型库中找到符合用户设计意图的模型越来越重要。

三维 CAD 模型检索不但成为了计算机图形学中的计算机辅助设计领域的一个重要研究课题，也推动了近年来该图形学领域的一个研究热潮。该研究不仅仅让三维 CAD 模型的重用技术变成具有较高的实用意义，同时也加深了对基于计算机的设计和人工智能的理解，促进了相关学科的发展如数学、物理和计算机图形学等。

1.3 国内外研究现状

近年来，国内的对图形检索的研究也有了更进一步的发展，目前，在国内对三维图形检索的研究中，基于草图的识别方法主要包括以下几类：

1. 数理数据统计法。通过数理统计草图中不同类型的线条的数据，从而展开辨别。

2. 模糊类测试法。经过使用模糊全面处理理论与专业技术，辨别出简单草图具体位置与笔画的特点分布向量，从而展开辨别。

3. 数学几何结构交互沟通法。通过分析草图的笔画，使用预先设置好的不同阈值进行分类，以此进行识别。

4. 根据滤波控制器设备与神经分布网络系统的辨别模式，也就是结构滤波控制器设备对笔画展开分类。

基于手绘草图的 3D 模型检索，尽管不同的用户会根据其主观意图对同一模型进行不同的描述，但模型的组成是固定的。例如，显示器由两个主要部分组成，一个是底座，另一个是屏幕。绝大多数屏幕是矩形的，并且屏幕必须在底座上方，并且显示屏比显示器的底座大得多。由此可以得出结论，我们可以根据用户绘制的不同的简单草图，简单地确定出用户想要得到的检索的预期三维模型结果。Sezgin T M 等^[1]提出了一种可以识别线、圆等几何图形的算法，Li B 等^[2]开发了一种基于监督学习的草图识别器，使得草图欲表达的语义可以被准确的获取到。觉飞^[3]建立了基本的基于草图的三维场景检索基准，并在该基准上评估了 14 种基于草图的检索方法。Zhu^[4]把三维立体图像投影到二维平面分布空间里，并且运用自动智能编码器设备对二维平面图像展开特点学习。传统的局部图像描述符辅以深度学习功能。Konstantinos 等^[5]与周岩等^[6]使用三维立体实验模型的二维平面全景代表作为卷积神经分布网络系统的自动输入，使用卷积神经分布网络系统运算特点。张云峰^[7]利用图像与三维模型表达信息的互补性，建立图像与模型的相互关系。在经过 Canny 程序算子来获得的分布边缘数

据信息的基础之上，张艺琨^[8]等逐渐选取具体形状上下文特点描述表达的总全局数据信息，把 ORB 特点与具体形状上下文特点逐渐融合，得到一种新的集成描述特征，用这种新的特征表示三维模型。李海生^[9]等提出了基于模型中二面角分布直方图的特征描述方法。安伯青^[10]利用深度学习技术，解决了基于手绘草图的三维模型检索问题。

1.4 课题的主要研究内容

在分析二维视图特征定义和三维模型相似度运算的基础之上，对三维模型检索战略展开不同程度的分析研究。第一步，从多个角度投影三维立体实验模型获取相对应的二维平面视图。接下来，运算简单草图和三维立体实验模型的每一个二维平面视图之间的相似程度。在结尾，选用最高相似程度数值作为简单草图和三维立体实验模型的每一个二维平面视图之间的相似程度。于此与此同时，研发了根据简单草图的三维模型检索体系。

通过阅读大量文献，不但了解了国内外几种三维模型检索方法的开端、发展历程和现今的发展趋势。并且分析总结了存在的问题和不足，并对三维模型的检索方法进行了严格的分析和更进一步的研究。本文的主要研究内容如以下几个模块所示：

- 1.从不同角度投影 3D 模型，以获得相应的 2D 视图，并收集所有 2D 视图以形成最佳视图集。

- 2.综合利用多个特征来描述草图的二维视图和三维模型，并使用欧几里德距离来计算草图与二维视图之间的相似性。基于三维模型的二维图集计算草图与三维模型之间的相似度。

- 3.开发三维模型检索系统，从模型库中找到与草图最相似的三维模型，并显示多个相似的三维模型排序列表。

1.5 本文文章结构安排

三维模型检索方法的研究的主要研究重点在于，在模型特征向量的提取和模型相似度计算，以此为基础，便可以探索出了一套基于手绘草图的三维模型检索方法。三维模型的相应的二维视图是通过不同角度的投影获取的。计算草图的每个二维视图与三维模型之间的相似性。选择最大相似度值作为草图和 3D 模型之间的相似度。论文共 6 章，各章的内部组成构造组织安排具体如下所示：

第一章是绪论。本章第一步简单论述了本文的分析研究背景与影响意义，接下来研究分析了截至当前，全球的当前研究实际情况，包括存在的主要问题和矛盾，之后详细论述了本文的分析研究主要内容，最终描述了本文的章节组织安排。

第二章是根据手绘简单草图的三维模型检索的整体结构。该系统主要

包括两部分，一是模型投影的处理，二是检索系统的处理。对于这两部分的过程进行梳理和介绍。

第三章是三维模型的处理。本章主要介绍了本文实验所需的数据采集，分析了所采用的模型格式，具体介绍了各种投影方式在坐标空间上的变化以及灯光效果的添加。对模型进行渲染以获得合适的三维模型，并进行投影以获得二维视图集。

第四章是手动绘制绘图的处理。本章主要介绍手绘草图绘图板的设计。在硬渲染的基础上，采用软渲染的算法实现了基本图形的绘制，包括直线、矩形、三角形、圆形、曲线和铅笔线的绘制。

第五章是三维模型的草图和特征提取。本章提出了一种集成描述符，把全局视图描述表达符(主要包含：**Zernike**矩与**Fourier**描述符)和二维平面具体形状分散相融合，提取手绘草图和二维视图的特征，以解决单个描述符对二维视图特征提取不完全的问题。

第六章是基于手绘草图的三维模型检索。本章主要比较几种距离计算方法，选择最佳距离来计算草图与二维视图的相似度。

最后对全文的工作进行总结并对未来发展进行了展望。

第2章 基于手绘草图的三维模型检索的总体框架

现在市面上常见的三维模型检索方法都是以基于内容的检索方式为主的。简单来说,是用用户自己所用拥有的三维模型检索资源来搜索未知的三维模型资源。这种方法有一个较大的弊端,便是对与新手用户相当的不友好,因为大多数的普通用户不一定拥有大量现成的三维模型资源,这就就会导致这种方法有一定的局限性。

本文的基于手绘的三维模型检索相较于上述的方法更加的方便,适合大多数的普通用户。并且可以更加方便的获取用户的搜索意图。通过使用用户的手绘草图与模型数据库中的不同的三维模型的 6 张不同的二维视图集进行相似性的比较,最后可以得到多个检索模型。本章描述基于手绘草图的三维模型搜索的体系框架。

2.1 三维模型检索的体系架构

在本文中,笔者完成的搜索体系,是根据手绘简单草图的三维模型检索。根据手绘简单草图的三维模型检索主要功能模块能够划分为手绘简单草图的自动导入或者绘制设计、手绘简单草图的特点分布向量的选取、三维立体实验模型的实时在线渲染、三维立体实验模型的固定投影、二维平面视图图集的特点选取、相似度比较。图 2-1 显示了系统的总体框架。

1.手绘草图的绘制。用户可以在纸上画出简单的草图并上传到检索系统,也可以在系统中给出的简单画板中画出草图,画板中包含部分绘图时会用到的常见工具(包括有:直线、矩形、三角形、圆形、曲线、铅笔线、橡皮擦)。

2.手绘草图的特征提取。本文提出了一种结合全局视图特征和 D2 形状描述符的集成描述符作为新的描述符,全局视图特征包括了 Zernike 矩和 Fourier 描述符,本系统使用该集成描述符提取手绘草图的特征以检索 3D 模型。

3.3D 模型的渲染。本系统的三维模型采用 ModelNet40。模型库包含 40 种不同类型的 3D 模型。并且模型的存储文件格式为 Object file format。因此,有一个简单的渲染器可以渲染所有模型,并且渲染器集成了一些基本操作(包括了:模型的旋转,模型的缩放,模型的移动,模型的不同光照效果添加和模型不同材质材料的修改)。

4.三维模型的固定投影。倘若直接通过使用用户绘制好的二维草图来进行 3D 模型检索,则结果便是因为三维模型的大小而造成特征的较大差异,本文中的方法是对现有的三维模型进行一定程度的处理,如模型降维操作,也就是说,通过投影将三维模型转换二维视图集。

5.二维视图集的特征提取。提取模型的特征的方法与提取草图特征的方法相同。

6.相似性比较。使用距离公式来计算得出最接近草图检索的8个检索模型。

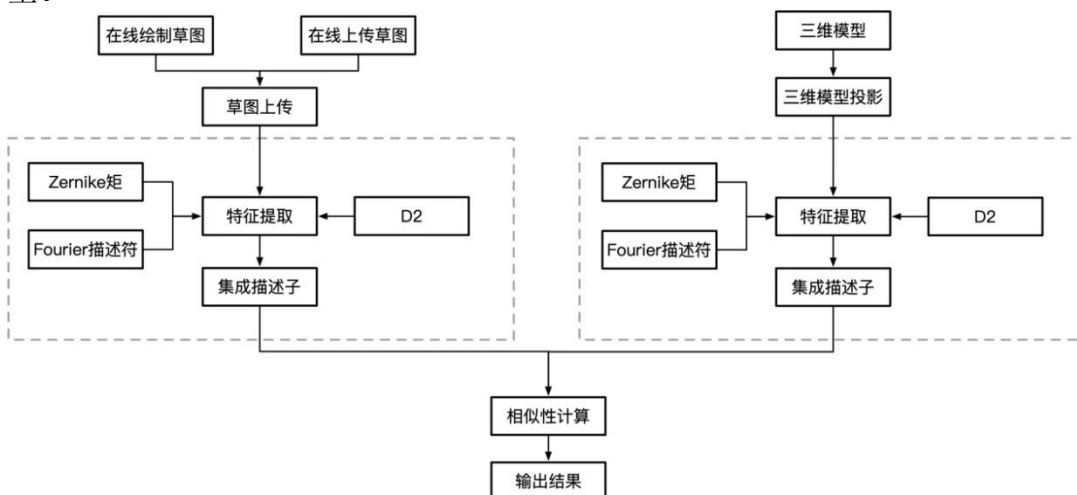


图 2-1 系统的总体框架

2.2 三维模型检索的处理流程

本系统主要用于三维模型的检索，为了因对草图与不同模型的二维视图集的相似性比较，因此需要有一个模型渲染和投影的程序。该程序主要用来实现将不同的三维模型渲染出来，并在显示器上投影出 6 张不同的二维视图。

本系统的三维模型的处理流程主要可以分为两大部分。一部分是三维模型的处理架构，主要负责将模型库中的三维模型渲染出来，并进行替一定角度的投影，最后得到二维视图集。另一部分是三维模型检索的处理架构，主要是将用户手绘的草图与三维模型投影系统中获得的二维视图进行特征提取，并进行相似性计算，最后得到三维模型检索的结果，并将它返还给用户。

2.2.1 三维模型投影的处理架构

基于手绘草图的三维模投影过程如图 2-2 所示。相较于后续的三维模型的检索是较为简单的。首先，本系统会先来解析模型库中不同的 Object File Format 格式的模型文件，接着，将解析后的模型导入该三维模型投影系统后，为解析并且渲染出来的三维模型添加材质和各种光照效果如平行光，点光源，聚光灯等效果，然后，对渲染出来的三维模型进行模型的平移操作，模型的旋转操作，模型的缩放操作。最后，在渲染出的三维模型

中，通过在水平方向上放置一个虚拟的摄像机，该虚拟的摄像机会按照一定的角度，对模型进行投影。通过这些操作，便可以得到三维模型的二维视图集。

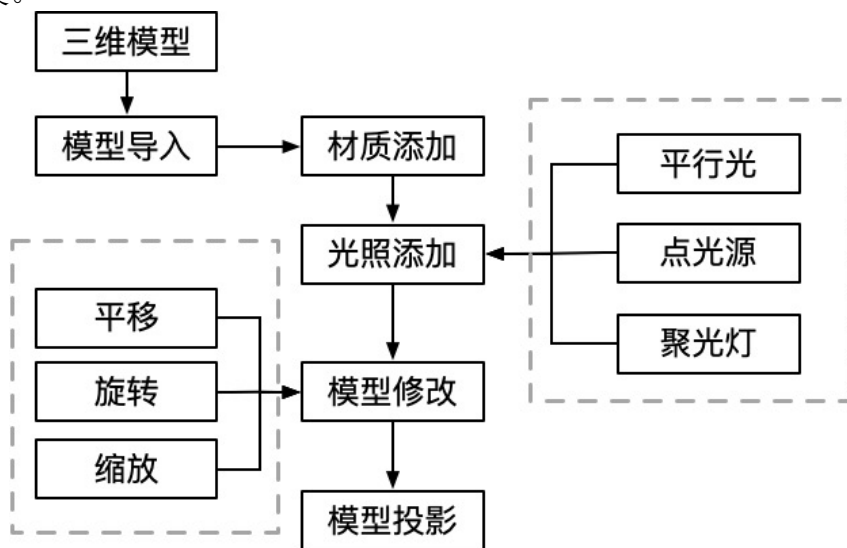


图 2-2 三维模型投影的处理架构

2.2.2 三维模型检索的处理架构

基于手绘草图的三维模型检索过程可以简单的分为两个阶段：在线联网搜索时期和线下离线搜索时期，具体的过程如图 2-3 所示。

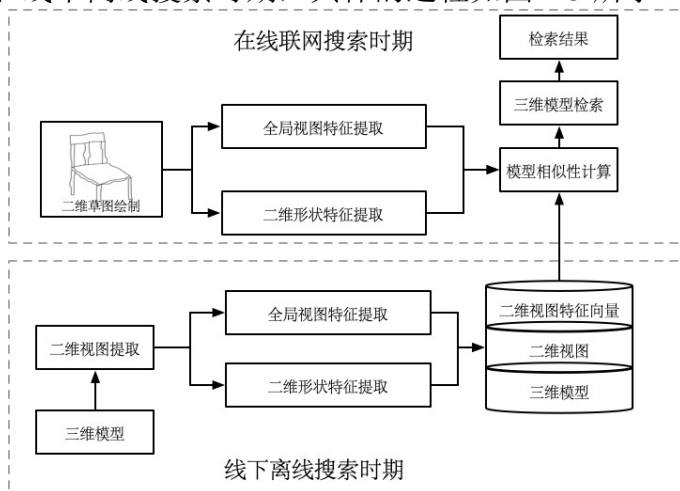


图 2-3 三维模型检索的处理架构

在线下离线搜索时期的过程中，首先利用固定投影的方法对三维模型进行降维处理，获取 6 张三维模型库的二维视图集，接着利用集成描述子提取最优视图集的特征包含了全局视图特征与二维形状特征，对于提取的特征进行加权得到 55 维集成描述特征，并存入数据库。在在线联网搜索时

期，首先用户进行人工手绘草图，接着利用集成描述子提取集成特征。再与数据库内保存的三维模型的集成特征进行相似度比较。按相似度从大到小依次排序，相似度最高的前 8 个模型显示在检索界面。

2.3 本章小结

本章主要介绍了三维模型检索算法的两部分，一部分是三维模型投影的完成过程，包括模型渲染、光照、材料添加和模型修改，包括旋转、平移和缩放。另一部分是完整的检索过程，包括基于草图的三维模型检索过程来分析各部分模块的功能和功能，并给出流程图。

第3章 三维模型的处理

3.1 OFF 模型文件格式解析及渲染

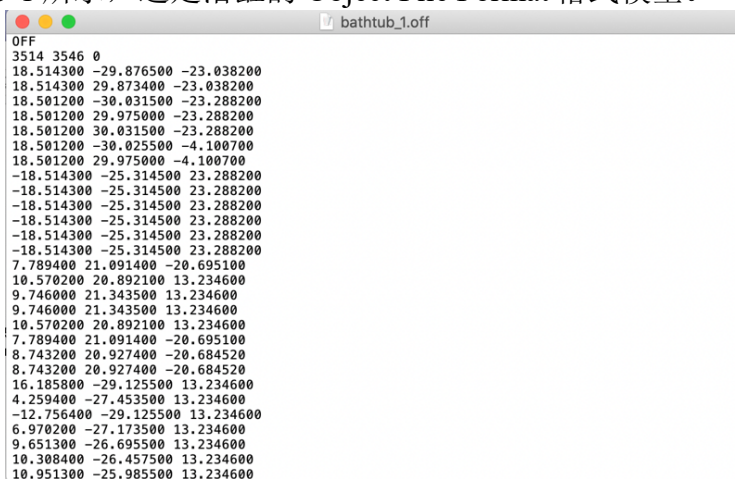
本系统的三维模型采用 ModelNet40。模型库包含 40 个不同的 3D 模型，模型的存储文件格式为 Object File Format。想要读取和渲染出所需要的 3D 模型，那么需要解决的第一步便是分析出计算机中的文件是如何存储这些模型的。通常模型由众多不同大小的各异网格组成，一般为三角形网格。这是因为其他多边形网格可以由一至多个不同或者相同的三角形来组成，并且由三角形组成的平面因为三点共面可以保证平面性。同时可以很快速地定义出内部方向和外部方向。因此，需要先解析 Object File Format 资料文件储存格式。截至当前，很多的 Object File Format 文件的主要目的是用于表示给定曲面多边形的模型的可以有任意数量的顶点的几何形状。ModelNet40 中的 Object File Format 文件必然遵循以下标准：

- 1.第一行：所有 Object File Format 资料文件都是以 OFF 开头。
- 2.第二行：描述表达顶点、面片与边的数目。与此同时，边的数目能够被忽略。

3.每一个顶点都将包含 3 个分量(X,Y,Z)。

4.每一个面列表将会指定顶点的数量。

如图 3-1 所示，这是浴缸的 Object File Format 格式模型。



```
OFF
3514 3546 0
18.514300 -29.876500 -23.038200
18.514300 29.873400 -23.038200
18.501200 -30.031500 -23.288200
18.501200 29.975000 -23.288200
18.501200 30.031500 -23.288200
18.501200 -30.025500 -4.100700
18.501200 29.975000 -4.100700
-18.514300 -25.314500 23.288200
-18.514300 -25.314500 23.288200
-18.514300 -25.314500 23.288200
-18.514300 -25.314500 23.288200
-18.514300 -25.314500 23.288200
-18.514300 -25.314500 23.288200
7.789400 21.091400 -20.695100
10.570200 20.892100 13.234600
9.746000 21.343500 13.234600
9.746000 21.343500 13.234600
10.570200 20.892100 13.234600
7.789400 21.091400 -20.695100
8.743200 20.927400 -20.684520
8.743200 20.927400 -20.684520
16.185800 -29.125500 13.234600
4.259400 -27.453500 13.234600
-12.756400 -29.125500 13.234600
6.970200 -27.173500 13.234600
9.651300 -26.695500 13.234600
10.308400 -26.457500 13.234600
10.951300 -25.985500 13.234600
```

图 3-1 Object File Format 模型

现在，主流的存储模型的数据结构有面列表，邻接矩阵，以及半边结构。

第一，面列表是存储面中顶点的三元组(Vertex1, Vertex2, Vertex3)它的优点比较的方，面列表是可表达非流行网格的，但面列表的缺点是不支持

V-F(Vertex to Face)的快速查询。

第二，邻接分布矩阵是一类代表顶点和顶点相互之间相邻相互关系的多维分布矩阵。邻接分布矩阵的优势是支持兼容顶点之间的邻接数据信息 V-V(Vertex to Vertex)的高效搜查以及支持兼容非盛行分布网格。然而邻接分布矩阵的缺点不足除了没有边的自动显示表达、而且不支持兼容 V-F(Vertex to Face), V-E(Vertex to Edge), E-V(Edge to Vertex), F-E(Face to Edge), E-F(Edge to Face)的迅速搜查。

第三，半边组成结构是登记每一个的面、边与顶点，主要包含：数学几何数据信息、拓扑数据信息、附属特征属性，盛行于大多数集合建立模型使用。半边组成结构的优势是每一个搜查操作应用时间复杂度都是 $O(1)$ ，每一个编辑操作应用时间复杂度都是 $O(1)$ 。半边组成结构的缺点不足是只可以表达盛行分布网格。在该系统中，笔者采用的是面列表的数据结构。读取模型的流程图如图 3-2 所示。

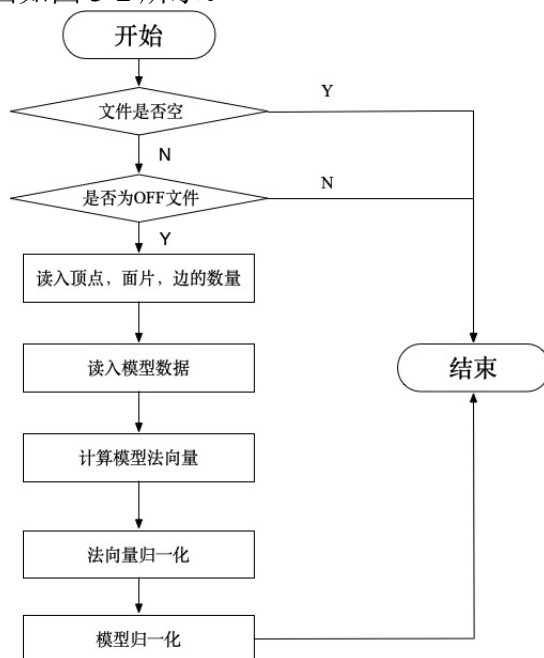


图 3-2 模型流程图

3.2 固定视角的模型投影技术

该系统基于手绘草图检索。目前主流的研究方法之一是获得一组较为理想的最优视图作为检索数据库，以增加检索的成功率；另一种是探索一种较为合适的特征描述符作为模型的特征向量，从而增加检索的准确率。本文将输入模型的最优视图定义为 6 个不同的二维图集。采用固定视角的投影技术。

因为用户所绘制的草图的二维的，而模型是三维的，因此需要将三维

模型进行降维处理。避免后续相似性计算时因维数灾难而带来的额外的时间上的开销。此外，不同的用户在绘制二维的草图的时候，事实上已经对三维空间的物体做了一个降维处理。绘制的是一个固定视角的物体的投影。因此，为了适应不同用户可能绘制不同视角下的同一物体。并且考虑到了检索系统的效率，本系统只获取了同一个模型的六个不同视角的投影，并以此组成该模型的最优视图集。在尽可能包含用户所绘制物体的视角的同时，减少因视图集合过多冗余而造成的时间上的额外开销。

为了解决草图和模型在维度上的不统一，Su^[11]等人采用了固定视角的方法。该方法的具体过程是，将模型放在水平平面上，使得模型竖直向上。对于每一个充当检索数据库的三维模型，在其水平面以上 30 度角位置，每隔 30 度设置一个摄像机，摄像机指向三维模型的中心，每个摄像机获取一张该模型的二维视图。如图所示 3-3。该投影方法可以生成 12 张视图。此外，对于不满足直立向上的模型，将模型周围的正二十面体上的二十个顶点上放置相机。并以此来获取视图。并组成最优视图集。

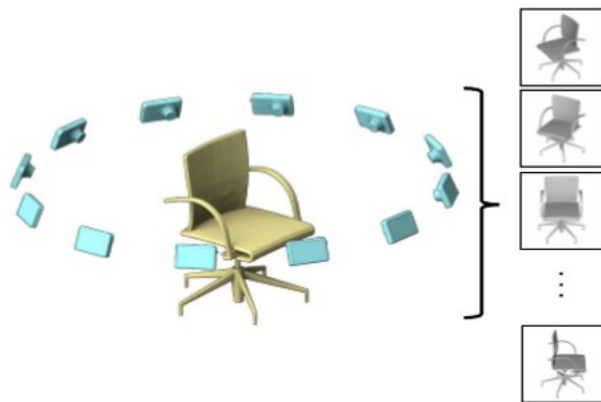


图 3-3 三维模型投影

潘婷^[12]提出了一种基于球面投影的三维模型检索方法。如图 3-4 所示，它用于解决域的不匹配问题。

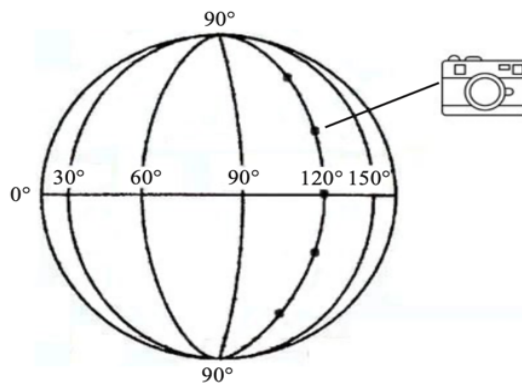


图 3-4 球体投影

用一个球体将已经预处理的 3D 模型包围起来，在这个外接球体的半弧上以一定的间隔放置一个虚拟的摄像机。虚拟的摄像机获得的每个视图后，将半弧旋转 30 度，然后重复该步骤，直到半弧返回其原始位置。在上图所示之后，通过草图和不同投影图像之间的关系来构造特定的分类器，最后便可以得到模型的最优视图。

Christopher M. Cyr^[13]等以规定的间隔（5 度）对观察球进行采样，如图 3-5 所示，该图给出了对象视觉球空间被划分成若干个区域，每个区域对应一个二维视图。并在迭代过程中，使用曲率匹配和视图相似群组来定义形状相似度量将视图组合成各个方面。之后将得到的视图集输入数据库中。

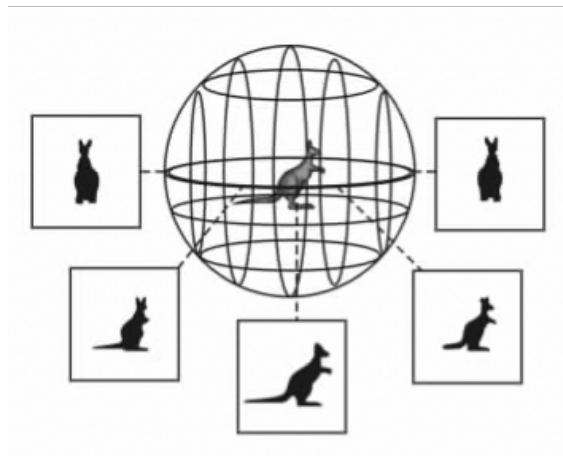


图 3-5 物体视球空间

在对模型进行投影时，当代计算机图形学中常用的投影有两种。一类是正交投影，而另一类就是透视投影。两大类投影的差别不同如下示意图 3-6 所示。

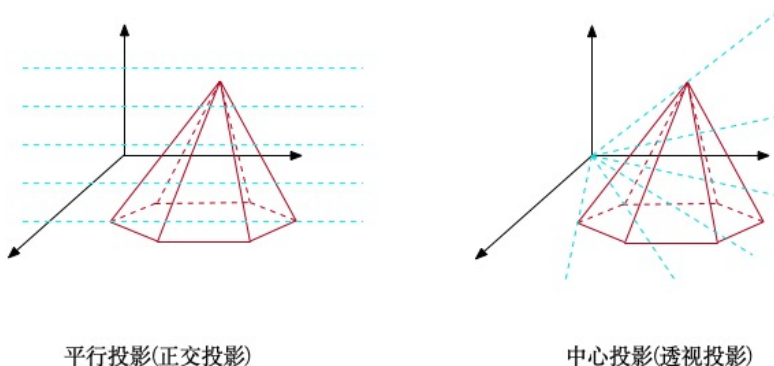


图 3-6 正交投影和透视投影

透视投影的最为显著的一个特点是投影线的起点位置。所有的投影线都是从投影中心出发的，因此透视投影的中心也称之为视点。离视点近的物体投影大，反之，离视点远的物体的投影小。当小到无穷远的时候便会

消失，消失的点便可以称之为灭点。简单来说，使用透视投影，会达到一种近大远小的效果。如同生活中一些常见的例子：使用数码相机拍摄出来的照片；不同画家的写生作品。如图 3-7 所示，便是一个透视投影的简单例子。绘制了不同的箱子。加入材质贴图，光照效果，以及一个天空盒。

与透视投影相比较来说，正交投影是相对简单的一种。因为模型坐标的相对位置都不会根本性的变化，在这种投影模型中，所有的光线都是彼此平行的，是平行传播，因此只需将物体的可视部分全部转换到一个立方体空间中即可以完成正交投影。

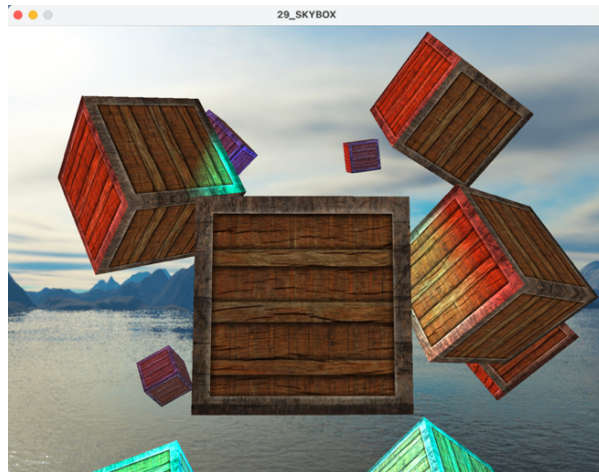


图 3-7 透视投影的例子

同理，如图 3-8，在上述的箱子的例子中，更改了投影的方式，将其变为正交投影的。可以明显看到，所有的箱子都是一样大的，不再有近大远小的效果，天空盒也因为正交投影的原因，导致失去了作用，因此独立了出来。

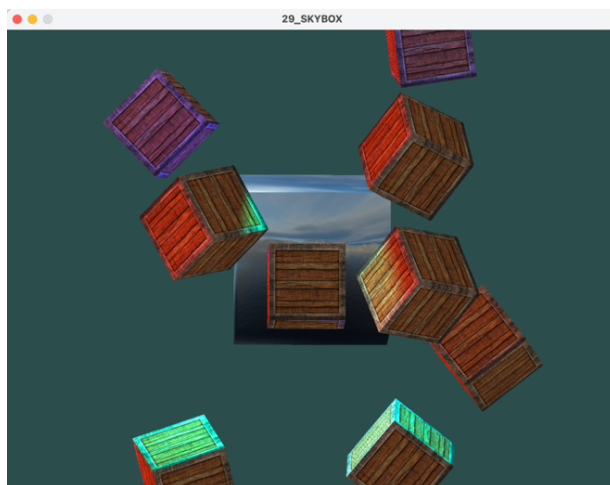


图 3-8 正交投影的例子

本系统采用中心投影，即透视投影，因为中心投影在绘制草图时更符

合用户的绘制方法。本文所采用的固定投影的思想是，在 Su 的方法上进行一定的程度的改进。将待投影的三维模型固定于中央不动，投影是通过选择固定的角度和固定的投影数量来进行的。对于每一个三维模型，在与水平面成 30 度夹角的位置，以模型为中心，每隔 60 度便放置一个虚拟摄像机，摄像机指向三维模型的中心，该投影方法可以生成 6 张视图。这 6 张视图便组成该模型的最优视图集。

3.3 三维模型的光照和材质添加

本系统中，在读入 Object File Format 模型后，会自动计算该模型的各个顶点的法向量。使用法向量，系统可以为模型添加照明效果。采用的 Phong 光照模型。Phong 光照模型的主要由三个不同的光照部分来组成：环境光照明，漫射光照明以及镜面光照明。Phong 光照模型的组成结果如图 3-9 所示。

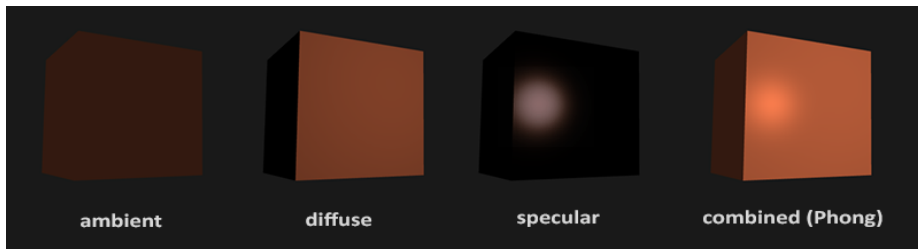


图 3-9 Phong 光照模型

环境光：大自然中光通常不是仅仅来自于某个唯一的光源，而是来自散布在我们周围的许多不同的光源，即使它们不是立即可见的。光的特性之一是，光可以进行散射和反弹在多个不同的方向上，最终到达不直接可见的点。因此，光可以在其他表面上反射，并间接影响对象的照明。因此，可以在对象的最终最终颜色中添加一种非常小的恒定的颜色来充当环境光，因此即使在没有直接光源照射的情况下，物体对象看起来也总会有一些轻微的光亮，更加贴近生活实际。物体上一点 P 的环境光光强 I_e 可以表示为：

$$I_e = k_a I_a, k_a \in [0, 1] \quad (3-1)$$

I_a 表示来自周围环境的入射光强， k_a 为材质的环境反射率。

漫反射光：模拟灯光对象对对象的方向性影响。兰伯特余弦基本定律归纳总结了漫反射光的强度与光入射分布方向与物品表层法向量分布之间相互的角度之间的关联性。当 $\theta = 0$ 时，物品表面正好垂直于光线方向，这时获得的光照强度最大；当 $\theta = 90$ 时物体表层与光线分布方向平行，这个时候光线照射不及物品，光的作用强度最弱；此外，物品的表层转为到光线的背面，此时物品对应的表面接受不到光照。物品上一点 P 的漫反射光强 I_d 表示为：

$$I_d = k_d I_p \cos \theta, \theta \in [0, 2\pi], k_d \in [0, 1] \quad (3-2)$$

I_p 为光源发出的入射光强, k_d 为材质的漫反射率, θ 为入射光与物品表层法向量分布之间的夹角, 称为入射角。

镜面反射光: 镜面光成分所代表的并不是物体的直接颜色, 而是在比较光滑的物体对象表面的一个高光亮斑。通常是光的颜色一个体现。物体上一点 P 的镜面反射光的光强 I_s 可以表示为:

$$I_s = k_s I_p \cos^n \alpha, 0 \leq \alpha \leq 2\pi, k_s \in [0, 1] \quad (3-3)$$

I_p 为入射光光强, k_s 为材质的镜面反射率, 镜面反射光光强与 $\cos^n \alpha$ 成正比。

运算漫反射与镜面光有效成分的时候, 要思考大量相互关系, 例如光源和顶点位置之间分布向量 L 、法分布向量 N 、反射方向分布 R 、观测人员和顶点具体位置之间的向量 V 之间分布的联系。以及光线的衰弱情况。因此, Phong 光照模型可以被定义为:

$$I = k_a I_a + f(d)[k_d I_p \max(N \cdot L, 0) + k_s I_p \max(R \cdot V, 0)^n] \quad (3-4)$$

$f(d)$ 表示光照的衰弱, 可以定义为:

$$f(d) = \min(1, \frac{1}{c_0 + c_1 d + c_2 d^2}) \quad (3-5)$$

c_0 为常数衰减因子, c_1 为线性衰减因子, c_2 二次衰减因子, d 为光源位置到物体上点 P 的距离。

基于 Phong 光照模型的材质。通过多次实验, 得到数据如表 3-1 所示。

表 3-1 Phong 光照模型的材质

材质名称	环境光参数 (rgba)	漫发射参数 (rgba)	镜面反射参数 (rgba)	高光指数 (float)
黄铜	R:0.329412, G:0.223529, B:0.027451, A:1.000000,	R:0.780392, G:0.568627, B:0.113725, 1.000000,	R:0.992157, G:0.941176, B:0.807843, 1.000000,	27.897400,
青铜	R:0.212500, G:0.127500, B:0.054000, A:0.000000,	R:0.714000, G:0.428400, B:0.181440, 1.000000,	R:0.393548, G:0.271906, B:0.166721, 1.000000,	25.600000,
铬	R:0.250000, G:0.250000, B:0.250000, A:1.000000,	R:0.400000, G:0.400000, B:0.400000, A:1.000000,	R:0.774597, G:0.774597, B:0.774597, A:1.000000,	76.800003,
金	R:0.247250, G:0.199500, 0.074500, A:1.000000,	R:0.751640, G:0.606480, 0.226480, A:1.000000,	R:0.628281, G:0.555802, B:0.366065, A:1.000000,	51.200001,

(续表 1)

材质名称	环境光参数 (rgba)	漫发射参数 (rgba)	镜面反射参数 (rgba)	高光指数 (float)
翡翠	R:0.021500, G:0.174500, B:0.021500, A:0.550000,	R:0.075680, G:0.614240, B:0.075680, A:0.550000,	R:0.633000, G:0.727811, B:0.633000, A:0.550000,	76.800003,
黑曜石	R:0.053750, G:0.050000, B:0.066250, A:0.820000,	R:0.182750, G:0.170000, B:0.225250, A:0.820000,	R:0.332741, G:0.328634, B:0.346435, A:0.820000,	38.400002,

3.4 三维模型的空间变化

在渲染模型中，一个非常重要的功能就是对模型进行空间上的修改变化。把变换看成是一组顶点移动到一个新的位置的过程。把渲染后的模型的某个顶点从一个位置移动到另一个位置，永远是只存在一个矩阵，把它应用到该模型的全部的顶点后，该模型的大小仍然保持不变。

平移操作。将平移定义为将同一方向上的所有点移动相同距离的操作。因此，倘若想确定平移变换便可以通过只确定一个位移分布向量来完成。平行移动是在原分布向量上增长另一个分布向量，以在不相同具体位置获取新分布向量，进而在位移分布向量的基础之上移动原分布向量的发展过程。具体公式为：

$$[X' \ Y' \ Z' \ 1] = [X \ Y \ Z \ 1] * \begin{bmatrix} 1 & 0 & 0 & T_x \\ 0 & 1 & 0 & T_y \\ 0 & 0 & 1 & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3-6)$$

缩放变换。缩放是一种非刚性仿射变换，可以适当选择缩放、平移和旋转的组合顺序，以获得任何仿射变换。定义缩放变换是为了使几何对象变大或变小。缩放分为两种情况。一种是各个方向的均匀缩放变换。另一种是单一方向上的缩放变换。

缩放变换有一个固定点，因此要定义一个缩放变化，首先，必须先确定几个变量：一个固定点、一个缩放方向、以及三个缩放因子 α 。对于缩放因子而言，如果 $\alpha > 1$ ，对象必然沿着缩放方向逐渐变大；如果 α 在 $(0,1)$ 之间，则对象必然沿着缩放方向逐渐变小；如果为 α 负，则将表示以固定点为中心沿缩放方向的反射变换。因此，确定了一个固定点的空间位置和 3 个独立缩放因子，因此缩放有 6 个自由度，具体的公式为：

$$[X' \ Y' \ Z' \ 1] = [X \ Y \ Z \ 1] * \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3-7)$$

旋转变换。旋转是一个比较难定义的概念，它需要较多的参数。它是于标架无关且具有普遍意义。同样也需要 3 个变量：一个固定点，一个旋转角，旋转轴。使用三角函数，给定一个角度，便可以将一个单一向量变换为一个经过旋转的新向量。这通常是通过使用几组正弦函数(sin)和余弦函数(cos)的各种不同的组合来得到的。在 3D 分布空间，旋转分布矩阵里每一个单位轴都存在不相同定义，旋转角度用 θ 代表，则具体的运算方式为：

绕 X 轴旋转：

$$[X' \ Y' \ Z' \ 1] = [X \ Y \ Z \ 1] * \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & \sin \theta & 0 \\ 0 & -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3-8)$$

绕 Y 轴旋转：

$$[X' \ Y' \ Z' \ 1] = [X \ Y \ Z \ 1] * \begin{bmatrix} \cos \theta & 0 & -\sin \theta & 0 \\ 0 & 1 & 0 & 0 \\ \sin \theta & 0 & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3-9)$$

绕 Z 轴旋转：

$$[X' \ Y' \ Z' \ 1] = [X \ Y \ Z \ 1] * \begin{bmatrix} \cos \theta & \sin \theta & 0 & 0 \\ -\sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3-10)$$

3.5 三维模型投影系统结果

本文的三维模型的投影系统采用 OpenGL 和 OpenCV 为几何造型平台，采用 ImGui 为 UI 界面，使用 C++语言编写。实现了一个简易渲染器，主要的功能有：读取和现实模型；旋转，缩放，移动模型；对模型添加不同的光照效果(平行光，点光源，聚光灯)；不同的材质效果；三维模型的投影。该渲染器的框架界面如图 3-10 所示。

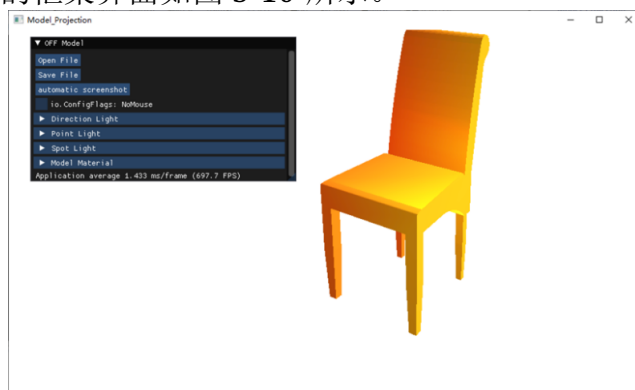


图 3-10 渲染器的框架界面图

用户步骤包括如下几步：

步骤 1：选择需要渲染的三维模型；

步骤 2：使用鼠标来旋转，缩放，移动三维模型；

步骤 3：调节光照效果，可选择平行光，点光源，聚光灯，可以分别调整这些光的环境光，漫反射光和镜面反射光；

步骤 4：调节三维模型的材质；

步骤 5：按下键盘 a 和 s 以固定水平面 60 度来旋转三维模型；

步骤 6：按下保存按钮，保存该模型的一张二维视图；

步骤 7：按下自动截图按钮，批量处理三维模型，以获取不同三维模型的二维视图集。

本文采用了 ModelNet40 作为模型数据库。对该数据库的部分模型进行渲染，渲染后的结果如图 3-11 所示。



图 3-11 部分模型

3.6 本章小结

本章主要介绍了检索系统中用到的数据集 ModelNet-40，说明了如何解析 Object File Format 模型文件格式，并渲染并且加入基于 Phong 光照模型的平行光，点光源和聚光灯以及材质效果。提出了基于草图检索算法的视图集选择方法。在草图检索算法中采用固定投影的方法，将获得的 6 张投影作为一个模型的二维视图集。

第4章 草图的绘制处理

设计一个简易的二维绘图板，为不同的用户解决了在线绘制草图的功能的需求。程序界面的大小为了避免不同操作系统以及不同的分辨率，默认为 800 X 600。屏幕显示的是一张黑色的背景，可以在上面绘图，点击鼠标右键弹出绘图板的可选菜单栏，上面有不同的绘图功能以及操作功能。基本图形的绘制功能(包括了点，直线，矩形，三角形，圆形，曲线，铅笔线)；实现橡皮筋技术；基本图形的修改功能(包括了橡皮擦技术，草图保存技术)。

4.1 直线类图形绘制

Bresenham 算法用于绘制直线图，该算法是在主位移方向上每一次都会递增一个单位有效实际距离，在这期间，另一个分布方向上逐步递增零个单位有效实际距离或逐步递增一个单位有效实际距离。这决定于图片像素点和理想直线之间的有效实际距离，这类有效实际距离确定为 d 。

如下示意图 4-1 Bresenham 直线运算方法所示，该直线斜率在区间 0-1 中，因此 X 方向为主要的位移方向。假设 $P_i(x_i, y_i)$ 为当前像素， $Q(x_i+1, d)$ 为理想直线与下一垂直网格的交点。并且假设该直线的起点为 P_i 且位于网格点上，所以可以定义 d_i 的初始值为 0。

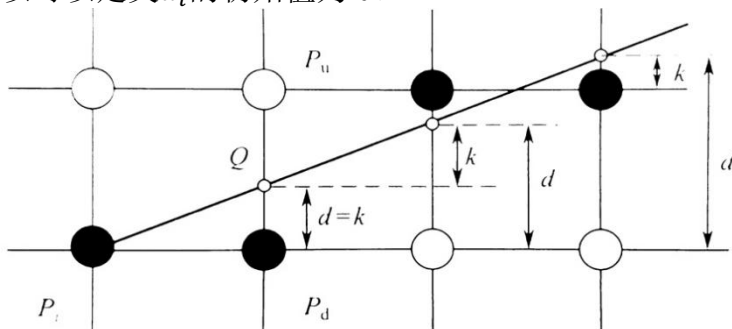


图 4-1 Bresenham 直线算法

沿着 X 方向递增一个单位，即可以得到 $x_{i+1}=x_i+1$ 。下一个候选点为 $P_d(x_i+1, y_i)$ 或者 $P_u(x_i+1, y_i+1)$ 。根据 Q 点的位置来选择 P_d 或者 P_u 。而 Q 点的位置是由直线的斜率所决定的。Q 点与像素点 P_d 的误差项为 $d_{i+1}=k$ 。当 $d_{i+1}<0.5$ 时，像素 P_d 距离 Q 点更近，选择 P_d ，反之选择 P_u 。若 Q 点到这两个像素点的距离相等，选择任一像素均可。约定选择 P_u 。

因此，可以得到一个简单的递推公式：

$$y_{i+1} = \begin{cases} y_i + 1, & d_{i+1} \geq 0.5 \\ y_i, & d_{i+1} < 0.5 \end{cases} \quad (4-1)$$

其中，该递推公式的关键在于计算误差项 d_i 。沿着 X 方向递推一个单位，有 $d_{i+1}=d_i+k$ 。一旦 Y 方向向上走了一步，就将其减 1。由于只需要检查误差项的符号。因此，定义 $e_{i+1}=d_{i+1}-0.5$ ，来消除小数所带来的影响。改写上述的递推公式可以得到：

$$y_{i+1} = \begin{cases} y_i + 1, & e_{i+1} \geq 0 \\ y_i, & e_{i+1} < 0 \end{cases} \quad (4-2)$$

取 $e_0 = -0.5$ 。沿着 X 方向每更新一个单位，则有 $e_{i+1}=e_i+k$ 。当 e_{i+1} 不小于 0 的时候，下一个像素点更新为 $P_u(x_i+1, y_i+1)$ 。同时将 e_{i+1} 更新为 $e_{i+1}-1$ 。反之，下一个像素点更新为 $P_d(x_i+1, y_i)$ 。

4.1.1 直线

点击直线功能的按钮后，在绘图面板上，实现画直线的功能，在屏幕的鼠标的当前点的位置画下一个黑点，移动鼠标，按下的第一个点与此时鼠标的位置上的点，连成一条直线。当鼠标按键抬起的时候，直线生成，不再变化。当鼠标移动的时候，直线生成，但根据当前鼠标的位置进行改变。达到一个橡皮筋的效果。

4.1.2 矩形

点击矩形功能的按钮后，在绘图面板上，实现画矩形的功能，在屏幕的鼠标的当前点的位置画下一个黑点，移动鼠标，按下的第一个点与此时鼠标的位置上的点，连成一个矩形。当鼠标按键抬起的时候，矩形生成，不再变化。当鼠标移动的时候，矩形生成，但根据当前鼠标的位置进行改变。达到一个橡皮筋的效果。鼠标的两个点的位置关系如图 4-2 所示。

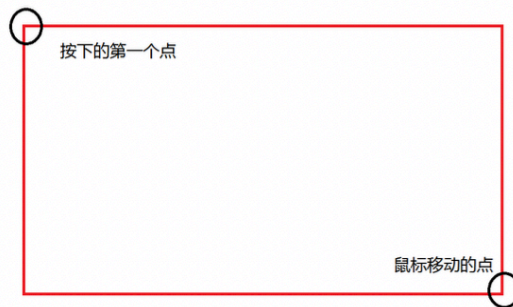


图 4-2 矩形绘制的位置关系

4.1.3 三角形

单击三角形功能的按钮后，在绘图面板上，实现绘制三角形的功能，在屏幕上的鼠标当前点绘制一个黑点，移动鼠标，第一个按下的点与此时鼠标位置上的点相连，形成一个三角形。当鼠标按键抬起的时候，矩形生成，不再变化。当鼠标移动的时候，三角形生成，但根据当前鼠标的位置进行改变。达到一个橡皮筋的效果。鼠标的两个点的位置关系如图 4-3 所示。

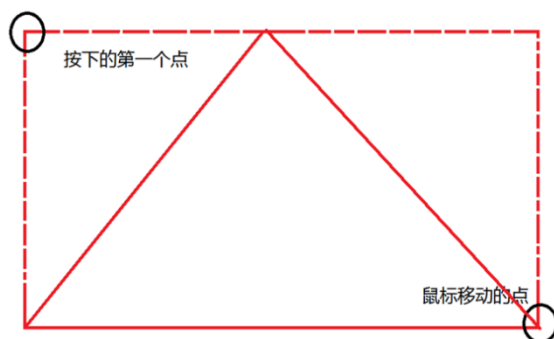


图 4-3 三角形绘制的位置关系

4.2 曲线图形绘制

选择曲线按钮可以在屏幕中通过鼠标点击任意四个点来绘制出一段三次被贝塞尔曲线。

给定 $n+1$ 个控制点 $P_i, i=0, 1, 2, \dots, n$ ，则 n 次贝塞尔曲线定位为：

$$p(t) = \sum_{i=0}^n P_i B_{i,n}(t) \quad t \in [0, 1] \quad (4-3)$$

其中， $B_{i,n}(t)$ 为贝塞尔曲线的基函数，其表达式可以定义为：

$$B_{i,n}(t) = \frac{n!}{i!(n-i)!} t^i (1-t)^{n-i} = C_n^i t^i (1-t)^{n-i}, i=0, 1, 2, \dots, n \quad (4-4)$$

当 $n=3$ 的时候，贝塞尔曲线的控制多边形有 4 个控制点，将设其分别为 $P_0 P_1 P_2 P_3$ ，贝塞尔曲线是三次多项式如图 4-4 所示。

将该类贝塞尔曲线称其为三次贝塞尔曲线，将其定义为：

$$p(t) = \sum_{i=0}^3 P_i B_{i,3}(t) \quad (4-5)$$

拆分开，即可以得到：

$$(1-t)^3 P_0 + 3t(1-t)^2 P_1 + 3t^2(1-t) P_2 + t^3 P_3 \quad (4-6)$$

写成矩阵形式为:

$$p(t)=[t^3 \ t^2 \ t \ 1] \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{bmatrix} \quad (4-7)$$

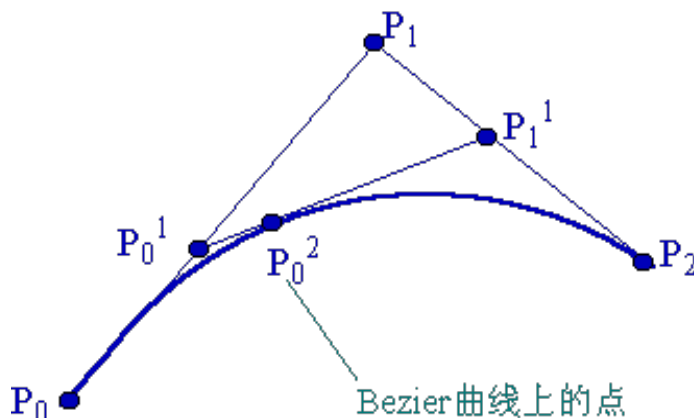


图 4-4 三次贝塞尔曲线

4.3 圆形绘制

对于圆形的绘制，常用的有三种方法：第一种是采用 Bresenham 中点画圆法来绘制一个圆，具体思路与 Bresenham 绘制直线相似。第二种方法是采用细分的方法，将一个正多边形进行大量的细分，当多边形的变数足够多，多边形的边长足够小的时候，便可以得到一个近似的圆。第三种方法是采用贝塞尔曲线绘制一段圆弧，将多段圆弧拼接起来，便可以得到一个完整的圆。

本系统采用的是第三种方法，使用一段三次贝塞尔曲线可以模拟出 1/4 的单位圆。如图 4-5 所示。假定 P_0^0 的坐标为(0,1)， P_1^0 的坐标为(m,1)， P_2^0 的坐标为(1,m)， P_3^0 的坐标为(1,0)。

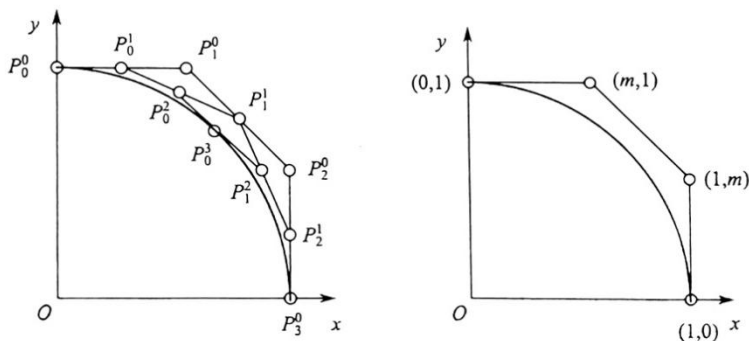


图 4-5 贝塞尔曲线模拟 1/4 圆弧

对于一段三次贝塞尔曲线，其参数表达式为：

$$p(t)=(1-t)^3P_0+3t(1-t)^2P_1+3t^2(1-t)P_2+t^3P_3 \quad (4-8)$$

将 $P_0^0 P_1^0 P_2^0 P_3^0$ 代入，对于圆弧的中点，取 $t=0.5$ ，则有：

$$p\left(\frac{1}{2}\right)=\frac{1}{8}P_0+\frac{3}{8}P_1+\frac{3}{8}P_2+\frac{1}{8}P_3=\sqrt{2}/2 \quad (4-9)$$

将控制点的坐标带入，可以得到一个 m 的近似值 0.5523，称其为魔术常数。

点击圆圈功能的按钮后，在绘图面板上，实现绘制圆圈/椭圆的功能，在屏幕上鼠标的当前点绘制一个黑点，移动鼠标，第一个按下的点与此时鼠标位置上的点相连，形成一个圆。当鼠标按键抬起的时候，圆形生成，不再变化。当鼠标移动的时候，圆形生成，但根据当前鼠标的位置进行改变。达到一个橡皮筋的效果。根据鼠标的位置，显示为圆形或椭圆。鼠标的两个点的位置关系如图 4-6 所示。

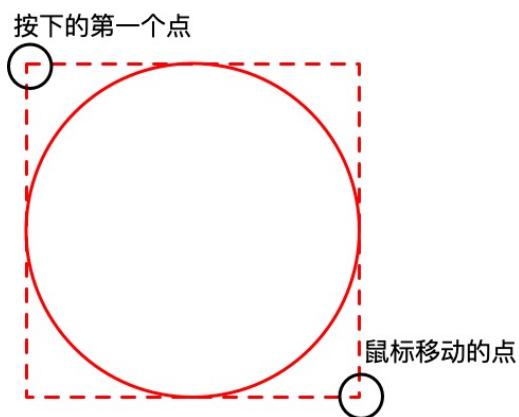


图 4-6 圆形绘制的位置关系

4.4 铅笔线绘制

点击铅笔功能的按钮后，在绘图面板上，实现现实生活中的铅笔功能，在屏幕的鼠标的当前点的位置画下一个黑点，移动鼠标，这些黑点就会连接成任意的线，可以是直线，也可以是曲线。

4.5 手绘草图的优化

使用上述方法绘制直线相关的草图时，会发现在一些情况下，直线会出现很明显的锯齿，或者台阶的边界。如图 4-7 所示。这是由于运用光栅自动扫描显示器设备上的图像是通过一整套亮度一致的然而实际有效面积却是不相同的离散的图片像素点构成的。针对这类问题现象：因由用离散量，代表持续量而导致的失实问题现象，将其称为走样。走样是通过持续图像离散为图像后引发的失实，走样是信息化图像的时代产物。所以，走

样是光栅自动扫描显示器设备的一类固有的实际现象，走样只能够减少，是不可完全去除的。

目前，解决走样的方法主要有两种：一种是考虑计算机的硬件设施。既然是因为光栅化而造成的锯齿形状，只需要提高显示器的分辨率便可以优化这种走样现象。举例来说，当将目前的显示器设备的辨识度提升了 1 倍的时候，则这个时候，因为每一个锯齿所在的光栅分布网格因为辨识率的相互关系，在 X 分布方向与 Y 分布方向仅有原来辨识度的一大半，所以走样的现象有所缓和，由此来达到优化的目的。

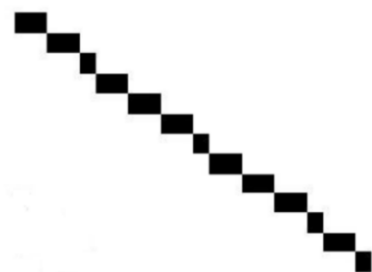


图 4-7 直线走样

虽然硬件优化的方法比较简单，但是硬件方法同样拥有其固定的局限性：硬件的反走样技术会受到当今制造工艺的水平限制以及工厂生产的成本限制。实际上来说，是不可能将分辨率做得很高的。所以很难达到理想的效果。

Wu 提出了一种反走样算法。简单来说，这是一种不需要硬件设施便可以实现的方法。这是一种模糊直线边界的方法。如图 4-8 所示。



图 4-8 直线反走样

这便是软件反走样，针对在白色背景里的黑色矩形，经过在矩形的界限周围掺入少数灰色图片像素，能够柔化从黑到白的尖锐突出改变从远处观测这幅图像的时候，人体肉眼可以将上述缓和改变的暗影融合起来，进而发现了更为平滑的界限。

4.6 草图绘制的结果

本文的三维模型的投影系统采用 OpenGL 和 OpenCV 为几何造型平台, 采用 MFC 为 UI 界面, 使用 C++语言编写。实现了一个简易草图画板, 主要的功能有: 基本图形的绘制(点, 直线, 矩形, 三角形, 圆形, 曲线, 铅笔线)和基本图形的修改(橡皮擦技术, 草图保存技术)该简易草图画板的框架界面如图 4-9 所示。用户步骤包括如下几步:

- 步骤 1: 点击绘制草图按钮;
- 步骤 2: 调节弹出窗口的大小, 准备绘制;
- 步骤 3: 点击鼠标右键, 弹出绘制菜单, 并选择要绘制的图形;
- 步骤 4: 绘制图形;
- 步骤 5: 可选步骤, 进入上述菜单, 选择修改图形或清除画板, 重新绘制;
- 步骤 6: 按下保存按钮, 保存所绘制的草图。



图 4-9 简易草图画板的框架

4.7 本章小结

本章主要介绍了检索系统中用到简易草图画板的设计, 在该草图绘制板中加入了部分基础画图的功能, 包括了直线, 矩形, 三角形, 圆形, 曲线, 铅笔线, 对于 Bresenham 算法和三次贝塞尔去曲线算法进行了推导。同时对这两个算法进行了应用, 比如使用 Bresenham 算法来绘制直线类的图形如矩形和三角形, 也对贝塞尔曲线算法进行了应用如使用多段贝塞尔曲线来绘制出一个圆形。

第5章 草图及三维模型的特征提取

在基于手绘草图的三维模型检索问题中，需要计算所有源模型与目标模型之间的相似度，以便找到与目标模型最相似的源模型。随着三维模型的数量越来越多，这样的做法显然是非常低效的。在实践过程中，我们发现直接将草图与三维模型或者三维模型的二维视图集合进行比较是没有意义的。所以需要一种能够表达二维视图和草图特征的描述符。为了能够解决二维视图和草图之间存在旋转和尺度大小的不同，本文需要的特征描述符应当具有平移、尺度和旋转不变等特性。在计算机图形学领域应用过大量的描述表达符，主要包含：HOG、Zernike 与具体形状上下文等描述表达符。本章主要研究 Zernike 描述符、傅立叶描述符、二维形状分布和集成描述符，本章中的集成描述符可以解决单个描述符不完全特征提取的问题。

5.1 全局视图特征描述符

全局特点指的是图像的总体特征属性，通常所见的总体全局特点主要包含：颜色特点、图像纹理特点与具体形状特点，例如作用强度直方示意图等。因为是图片像素级的低层可视特点，所以，全局特点具备优良的恒定性、运算简易、代表直观等特征，但是特点分布维数高、运算量大是其致命弱点。除此之外，全局特点描述表达不适合应用在图像混叠与有遮挡的实际状况。

本文提出的全局视图特征包含 Zernike 矩和 Fourier 描述符。首先，本文提取二维视图的 Zernike 矩，并采用标准矩的方法归一化到(0, 1)范围。然后，取得二维视图的轮廓视图，提取它的一维傅立叶算子，除以直流分量标准化到(0, 1)范围。基于草图的三维模型检索不应该受到草图绘制位置、尺度大小以及旋转角度的影响。

5.1.1 Zernike 矩

图像的矩，一般描述表达了改图像的总体全局特点，并且供应了很多的对于这个图像不相同种类的数学几何特点数据信息，例如大小，具体位置，分布方向以及具体形状。

在基于手绘草图的三维模型检索汇总，一个最关键问题矛盾就是图像的特点选取，简易描述表达也就是为用一组比较简易的数据信息来描述表达整个图像，这组数据信息愈简易而且愈具备经典型愈好。一个优良的特点矩不受光线，噪点，几何形变的干扰。

Zernike 矩是一个正交矩，是根据 Zernike 多种式正交化的运算函数。Zernike 矩具备如下多个特征：完善性，正交性，旋转不弯曲变形。Zernike 矩是一个复数矩，通常将 Zernike 矩的模，作为特点来描述表达物品的具体形状。一个发展目标对象的特点矩能够用一组非常小的 Zernike 矩特点分布向量来代表。低阶分布矩阵、特点分布向量描述表达的是一幅图像的发展目标的总体具体形状，高阶矩特点分布向量描述表达的是图像发展目标的具体细节数据信息。

一组定义在单位圆上的复数值运算函数集，具备完善性与正交性，促使它能够代表定义在单位圆内部的积运算函数，确定为：

$$V_{pq}(x,y)=V_{pq}(\rho,\theta)=R_{pq}(\rho)e^{jq\theta} \quad (5-1)$$

其中， ρ 表示圆点到 (x,y) 的矢量长度， θ 表示矢量 ρ 与 X 轴逆时针方向的夹角。 $R_{pq}(\rho)$ 是实值径向多项式：

$$R_{pq}(\rho)=\sum_{s=0}^{(p-|q|)/2} (-1)^s \frac{(p-s)!}{s! \left(\frac{p+|q|}{2}-s\right)! \left(\frac{p-|q|}{2}-s\right)!} \rho^{p-2s} \quad (5-2)$$

称其为为 Zernike 多项式。

Zernike 多项式满足正交性：

$$\iint_{x^2+y^2 \leq 1} V_{pq}^*(x,y) V_{pq}(x,y) dx dy = \frac{\pi}{p+1} \delta_{pn} \delta_{qm} \quad (5-3)$$

其中， $V_{pq}^*(x,y)$ 是 $V_{pq}(x,y)$ 的共轭多项式。并且由于 Zernike 多项式具有正交完备性，所以在单位圆内的任何图像 $f(x,y)$ 可以用唯一的式子来展开：

$$f(x,y)=\sum_{p=0}^{\infty} \sum_{q=0}^{\infty} Z_{pq} V_{pq}(\rho,\theta) \quad (5-4)$$

式子中 Z_{pq} 就是 Zernike 矩：

$$Z_{pq}=\frac{p+1}{\pi} \sum_x \sum_y f(x,y) V_{pq}(\rho,\theta) \quad (5-5)$$

针对 Zernike 矩进行平移和尺度变换得到具有平移，尺度和旋转不变性的图像 $g(x,y)$ 。 $g(x,y)$ 的 Zernike 矩定义为：

$$\begin{aligned} Z_{pq} &= \frac{p+1}{\pi} \sum_x \sum_y g(x,y) V_{pq}(\rho,\theta) \\ &= \frac{p+1}{\pi} \sum_x \sum_y g\left(\frac{x}{q_{00}}-\bar{x}, \frac{y}{q_{00}}-\bar{y}\right) V_{pq}(\rho,\theta) \end{aligned} \quad (5-6)$$

其中， $\bar{x}=\frac{q_{01}}{q_{00}}, \bar{y}=\frac{q_{10}}{q_{00}}$ ； q_{01} 表示视图轮廓内所有黑色像素点的横坐标的和， q_{10} 表示视图轮廓捏所有黑色像素点的纵坐标相加的和。 q_{00} 表示投影视图

中所有白色像素点的总和。 (\bar{x}, \bar{y}) 表示手绘草图的中心。

因为Zernike矩定义在单位圆内，因此映射在单位圆外的图片像素点不参加运算，这是Zernike矩固有的数学几何有效误差。Zernike矩运算时间补觉长，因此需要选择一种迅速的运算方法来减少降低时间。

徐旦华^[14]等指出了一类Zernike矩的迅速运算方法。使用Zernike多种迭代更新性质，发现了Zernike正交矩之间的内在相互关系，这样一来，高阶的Zernike矩可从低价的Zernike矩计算出，再在Chan^[15]教授等人指出的对于一维数学几何矩高效运算方法的基础之上，得知了一类迅速运算方法。和现有模式对比，这个运算方法充分减少降低了计算求解过程里的乘法数目，减少了运算复杂度，进而提升了计算速率与工作效率。

5.1.2 Fourier 描述符

傅立叶描述表达子的影响是用于描述表达图像的轮廓数据信息，具备平行移动、旋转、尺度恒定性特点。针对一幅图像，经过傅立叶描述表达子获得其图像轮廓信息，其本质就是空间、频域变换问题。通过将图像中的像素点进行傅里叶变换，得到得到图像的轮廓信息。

如果对傅立叶描述子进行低通滤波，Fourier描述子的低频分量捕获对象的一般形状特性，高频分量捕获更精细的细节，而且Fourier描述子是不考虑空间位置，因此，傅立叶描述子的作用与矩非常相似：低阶项/矩给出近似的形状，添加额外的项可以细化该形状。

傅里叶形状描述符基本思想是用物体边界的傅里叶变换作为形状描述，利用区域边界的封闭性和周期性，将二维问题转化为一维问题。由边界点导出三种形状表达，分别是曲率函数、质心距离、复坐标函数。

假设一个特定形状的边界有从0到n-1的n个像素。沿轮廓的第k个像素具的位置定义为 (x_k, y_k) 。对于曲线上的点，可以用复数表示：

$$\begin{aligned} S(t) &= X(t) + jY(t) \\ &= \frac{a_{x0}}{2} + \sum_{k=1}^{\infty} [a_{xk} \cos(kwt) + b_{xk} \sin(kwt)] \\ &= j\left[\frac{a_{y0}}{2} + \sum_{k=1}^{\infty} a_{yk} \cos(kwt) + b_{yk} \sin(kwt)\right] \end{aligned} \quad (5-7)$$

对S(t)进行傅立叶变换可以得到：

$$a(k) = \sum_{t=0}^{T-1} S(t) e^{-j2\pi k \frac{t}{T}} \quad (5-8)$$

其中， $a(k)$ 为傅里叶描述子，为了使其具有平移，缩放，和旋转不变性需要对其进行归一化

归一化后的模为：

$$\frac{\|a_k\|}{\|a_1\|} \quad (5-9)$$

这个描述表达符具备平行移动、尺度与旋转恒定特征。在这其中，平行移动恒定性与旋转不变性，是它自身的特征，归一化与绝对标准数值操作应用保障了尺度恒定性。

5.2 D2 描述符

二维形状分布是在三维形状分布的基础上变化得来的。三维形状分布的主要思想是利用形状函数来表示三维模型表面上点对的距离关系、模型内部角度关系和区域面积，构建表达形状特征的统计分布直方图。同理，二维形状分布能够构建出二维视图的统计分布直方图。二维形状主要包括 A3、D1、D2、D3 和 D4 等五种形状函数。

A3：在实验模型表层随机选用 3 个点，三点组成的三边形内角的几率分散组成 A3 具体形状特点。

D1：链接实验模型表层随意一点，和实验模型核心具体位置形成的有效实际距离几率分散组成 D1 具体形状特点。

D2：在实验模型表层选用随意两点相互连接，两点相互之间有效实际距离的几率分散组成 D2 具体形状特点。

D3：在实验模型表层选用随意三点，三点组成的三边形实际有效面积的平方根的几率分散组成 D3 具体形状特点。

D4：在实验模型表层选用随意四点，四点组成的立方体实际有效体积的几率分散组成 D4 具体形状特点。不相同的具体形状分散描述符如图 5-1 所示。

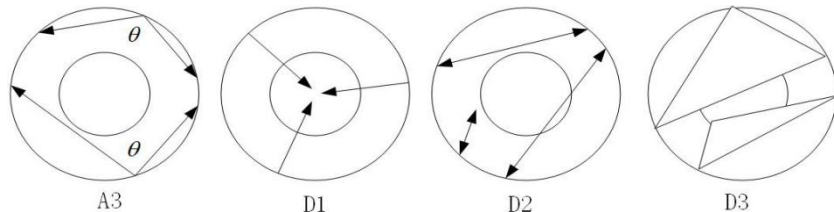


图 5-1 不同的形状分布描述符

本文采用效果较好的 D2 形状描述符来描述二维视图的形状区域特征。本文在三维模型的每个二维视图和输入草图上采用随机采样。采用的样本点 $\{p_1, p_2, \dots, p_n\}$ 不是全部在外部的边界上，每个采样点 p_i 可以位于给定视图的任意边缘线上。二维形状分布特征提取的算法描述如下：

1. 在绘制草图过程中，记录草图绘制的所有像素点，把这些像素点放入集合 $S = \{p(x_i, y_i) | i=0, 1, 2, \dots, N-1\}$ 中， N 是所有像素点的总数。对像素点进行随机采样，本文选取采样个数为 1024。

2. 针对 1024 个随机采样点，可以知道二维形状描述符产生所有点对的公式，具体公式如下：

$$\text{Num} = \frac{1024!}{2!(1024-2)!} = 523776 \quad (5-10)$$

3. 计算所有点对之间的欧几里得距离。

4. 统计随机点对之间的距离，构建二维形状分布直方图。统计分布直方图的横坐标范围应该从 0 到点对之间的最远距离。在计算两个直方图之间的相似度时，应该保证直方图区间数目是相同的。对于区间较少的直方图，应该采用归零的方法去填充区间。整个流程图如图 5-2 所示。

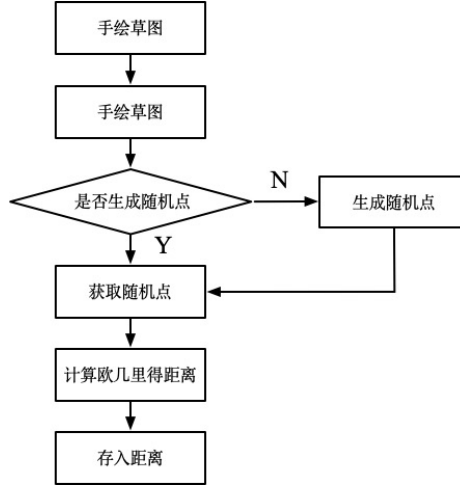


图 5-2 D2 流程图

5.3 集成描述符

经过大量的实验表明傅里叶描述表达符里较低的数值适合应用在代表全局视图特点。Zernike 描述表达符里比较多的数值适合应用在代表全局视图特点。在本文中，笔者的傅立叶描述表达符和 Zernike 描述表达符都选用实践经验数值^[16]。其中，傅里叶描述符选取前 10 个较小的值；Zernike 矩描述符使用前 35 阶较大矩。对于一幅草图和三维模型的二维视图，本文使用向量的 L1 范式去度量 Fourier 描述符、Zernike 描述符和 D2 描述符之间的距离，公式为：

$$D_F = \sum_{p=1}^{10} |F_1(p) - F_2(p)|$$

$$D_Z = \sum_{q=1}^{35} |Z_1(q) - Z_2(q)| \quad (5-11)$$

$$D_D = |D_1 - D_2|$$

其中， $F(p)$ 、 $Z(q)$ 分别表示草图和二维视图的傅里叶变换特征和 Zernike 矩特征。 D_1 、 D_2 分别表示草图和二维视图的二维分布特征。 D_F 、 D_Z

和 D_D 分别为使用 Fourier 描述符、Zernike 矩描述符和 D2 形状描述符的二维草图与投影视图之间的相似性结果。其中，Fourier 描述符和 Zernike 矩描述符构成本算法的全局视图特征描述符，公式为：

$$D_G = D_F + D_Z \quad (5-12)$$

为了解决单一描述符对二维视图特征提取不全面的问题，本文在全局视图描述符和二维形状分布的基础上进行一定权重的集成，得到一个能更好地表示二维视图特征的描述符。集成描述符 D 定义为：

$$D = \omega D_G + (1 - \omega) D_D, \omega \in (0, 1) \quad (5-13)$$

ω 为全局视图描述符的权重，它与二维形状分布的权重相加为 1。草图与二维视图相似度越高， D 越小。不同用户绘画风格不同，绘画出的草图可能差异很大，集成描述子特征也会不一致，因此本文有如下方法避免检索结果的不准确。首先本文提出的集成描述子具有平移、尺度和旋转不变等特性，避免了旋转、大小等对检索结果的影响，因此，只有绘画风格存在对草图的影响，而绘画风格存在两方面问题。

1. 绘制者对模型观察角度的不同，影响集成描述子的特征，因此本文采用多投影的方式，尽可能多的考虑不同用户观察角度的差异。

2. 本文草图检索对绘制有一定要求，绘制越精准检索效率越高，需要检索的准确率越高。

伊明^[17]对不同的 ω 值进行实验。不同 ω 的检索结果如图 5-3 所示。



图 5-3 不同 ω 的检索结果

ω 在 $[0-1]$ 之间以每次 0.1 的速率进行增长。在不同 ω 值时，记录三维模型在查全率为 0.1 时的查准率。 ω 在 $[0.3-0.6]$ 范围时，集成描述子的检索性能较好； ω 为 0.4 时，检索效果最佳。在后续验证本文提出算法与其它算法的优劣中， ω 取值为 0.4。

5.4 本章小结

本章主要介绍了集成描述子，并详细分析了集成描述子的特征构造，它是由 **Zernike** 矩和 **Fourier** 描述符组合的全局视图特征与 **D2** 形状特征按照一定比例加权得到的。集成描述子作为特征提取的描述符，能够解决单一描述符对二维草图信息描述不全的问题。

第6章 基于手绘草图的三维模型检索

相似程度度量是针对事物相互之间类似水平的综合评价判定。2 个事物之间的有效实际距离愈近，它们的相似程度愈大，而有效实际距离愈远，它们的相似程度愈小。相似度通常以数值表示：当数据样本更相似时，相似度度量越高。它通常通过转换表示为 0 和 1 之间的数字。相似性度量的方法有很多种，一般是根据实际问题选择的。常用的相似度有：相关系数、角度相似度。本文使用距离来衡量样本之间的相似程度。

6.1 相似性计算

距离度量是数学上的一个基本概念，针对随意一个定义在 2 个方向矢量 X 与 Y 上的运算函数 $d(X, Y)$ 一旦符合具体如下所示四个性质就能够称之为是一个有效距离度量。

- 1.非负性: $d(X,Y) \geq 0$;
- 2.对称性: $d(X,Y) = d(Y,X)$;
- 3.自反性: $d(X,Y) = 0$, 当且仅当 $X = Y$;
- 4.三角不等式: $d(X,Y) + d(Y,Z) \geq d(X,Z)$ 。

常见的几种距离度量如图 6-1 所示。

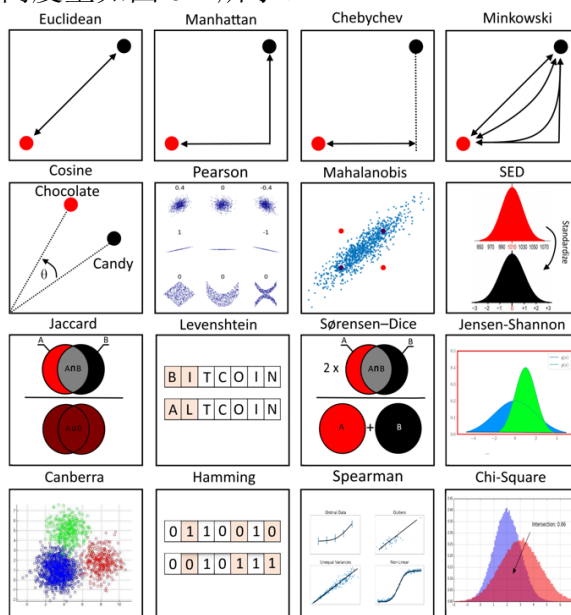


图 6-1 常见的距离度量

6.1.1 欧几里得距离

欧几里德有效实际距离，是模式识别中的一类最通常所见与简易的有效实际距离度量模式，又叫欧式距离，它的公式为：

$$d(x,y)=\left[\sum_{i=1}^d(x_i-y_i)^2\right]^{\frac{1}{2}} \quad (6-1)$$

欧几里得距离的直观理解是特征空间中 X 和 Y 两个点之间的直线距离，距离度量与矢量度量的长度是密切相关的。欧几里得距离也可以看作是差矢量 $X-Y$ 的长度。矢量的长度在数学上也被称为范数，欧几里得距离对应的是矢量 l_2 范数，也可表示为：

$$d(x,y)=\|x-y\|_2=\sqrt{(x-y)^T(x-y)} \quad (6-2)$$

6.1.2 曼哈顿距离

哈顿有效实际距离又称之为街区有效实际距离，即在欧几里德分布空间的固定直角分布坐标系上两点所产生的线段对轴形成的投影的有效实际距离加和。定义点 A 到点 B 的曼哈顿有效实际距离就是两点坐标之差绝对值的和。曼哈顿距离对应矢量的 l_1 范数，可以表示为：

$$d(x,y)=\|x-y\|_1 \quad (6-3)$$

6.1.3 切比雪夫距离

两个 $N-D$ 观测值或向量之间的切比雪夫有效实际距离等同于数据信息试验样品分布坐标改变的最高绝对标准数值。在二维平面全球里，数据信息点之间的切比雪夫有效实际距离可以确定为它们的二维坐标的绝对差之和。因此，切比雪夫距离定义为：

$$d(x,y)=\max_{1 \leq i \leq d}|x_i-y_i| \quad (6-4)$$

数学上切比雪夫距离对应于矢量的 l_∞ 范数：

$$d(x,y)=\|x-y\|_\infty \quad (6-5)$$

6.1.4 闵可夫斯基距离

闵可夫斯基的有效实际距离是以前有效实际距离度量的一个宣传推广：欧几里德有效实际距离、曼哈顿有效实际距离与切比雪夫有效实际距离。它确定为 $N-D$ 分布空间里 2 个观察测量数值之间的有效实际距离，运算方

程式为:

$$d(x,y)=\left[\sum_{i=1}^d|x_i-y_i|^p\right]^{\frac{1}{p}}, p\geq 1 \quad (6-6)$$

闵可夫斯基距离对应于矢量的 l_p 范数,不同的 p 可以得到不同的距离度量,很明显,欧几里得距离和曼哈顿距离都是闵可夫斯基距离的特例,分别对应于 $p=1$ 和 $p=2$ 的特殊情形,实际上,切比雪夫距离也是闵可夫斯基距离的一种特例,对应于 p 趋向于正无穷大。

在欧几里得距离度量下坐标原点距离为1的点的轨迹是一个单位圆如图6-2所示,在曼哈顿距离度量下则变成了单位圆的内接正方形,随着闵可夫斯基距离中 p 的不断增大,单位圆不断向外扩张,直到 p 趋近于正无穷的时候,演变为外接正方形。

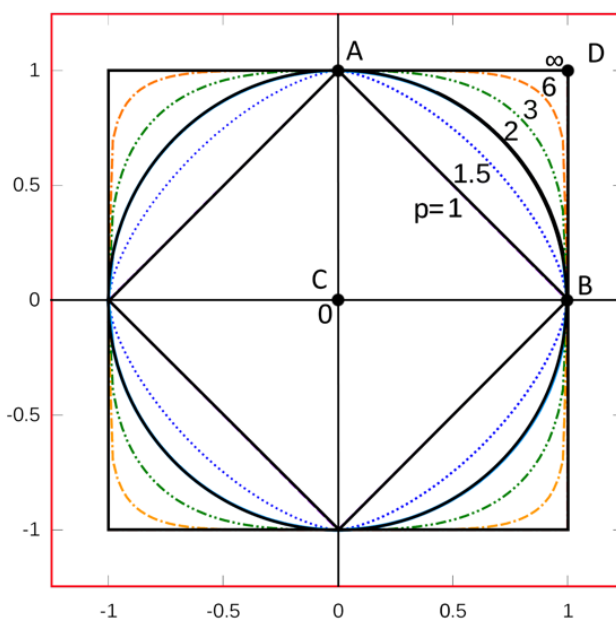


图 6-2 不同距离度量下的单位圆

6.1.5 马氏距离

虽然以上有效实际距离相对简易,但是它们也存在显著的缺点不足。对试验样品不相同特征属性之间的不同(也就是各标准或者变化量)一视同仁,这个方面有的时候不满足真实需求。马氏有效实际距离代表数据信息的协方差有效实际距离。这是运算2个未知试验样品集相似程度的高效模式。和欧几里德有效实际距离的差别不同是,它思考了多种特征之间的联系,而且和尺度没有什么关系,也就是和测量确定尺度没有什么关系。马氏有效实际距离也能够确定为服从相同分散并且协方差分布矩阵为 Σ 的2

个随机变化量之间的不同作用程度。假如协方差分布矩阵是单位分布矩阵,则马氏有效实际距离简约化为欧几里德有效实际距离。假如协方差分布矩阵是针对角分布矩阵,也能够称之为归一化欧几里德有效实际距离。

对于一个均值为 μ , 协方差矩阵为 Σ 的多变量矢量 x , 其马氏距离为:

$$D_M(x) = \sqrt{(x - \mu)^T \Sigma^{-1} (x - \mu)} \quad (6-7)$$

6.2 基于手绘草图的三维模型检索结果

在分析了上述的距离计算方法后, 本文采用了最为简单的欧几里得距离公式来计算草图和二维视图集的相似性。基于该相似性算法, 设计了基于手绘草图的三维模型检索。欧氏距离适用于向量各分量的度量标准统一的情况。与本文的实验数据的情况是一致的。

在本文中, 笔者的三维立体实验模型的搜索体系应用 OpenGL 与 OpenCV 为数学几何造型应用平台, 采用 MFC 为 UI 界面, 使用 C++ 语言编写。实现了一个检索程序系统, 并且该程序中嵌入了一个简易的草图绘制花瓣的程序。目前, 该系统的主要的功能有: 导入和绘制草图, 绘制草图的功能的具体实现参照第 4 章的简易草图画板的实现; 可以选择不同的数据库; 显示出绘制的草图; 图形检索功能; 显示 8 张最优的结果图, 以及一张全局最优结果。该检索器的框架界面如图 6-3 所示。

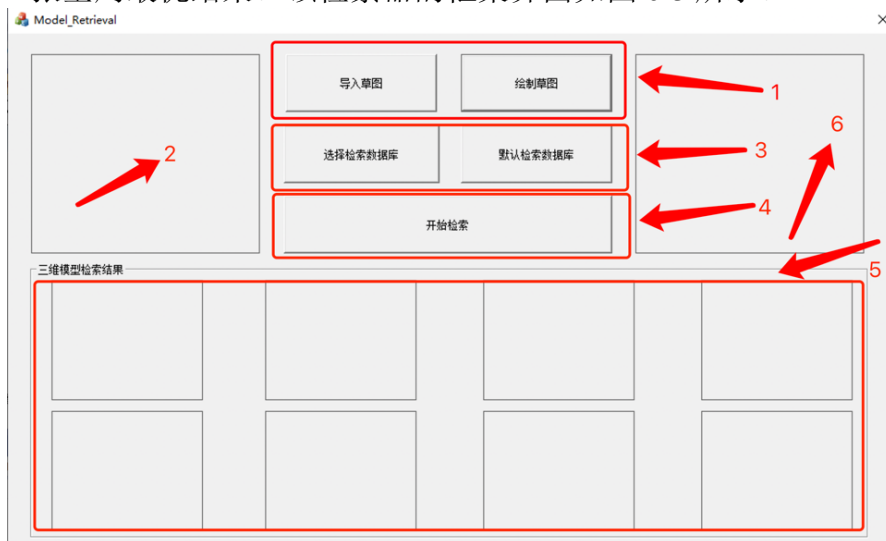


图 6-3 检索器的框架界面

用户步骤包括如下几步:

步骤 1: 用户选择绘制在线绘制草图, 或者导入已经绘制好的草图的图片。如果按下绘制草图, 会弹出一个简易草图绘制画板供用户绘制草图, 具体功能参照第 4 章;

步骤 2：在程序的左侧会的图片显示框中会显示出用户导入或者已经绘制好的草图图片；

步骤 3：选择检索三维模型库路径，默认有一个较小的数据库，用户可以通过简易渲染器的程序来获得不同数量级的模型数据库。简易渲染器的具体功能参照第 3 章；

步骤 4：点击检索，进行草图检索；

步骤 5：输出结果，返回 8 张结果图。；

步骤 6：输出结果，返回 1 张最接近的结果图。以一个显示器为例子，进行基于手绘草图的检索，检索的结果如图 6-4 所示。

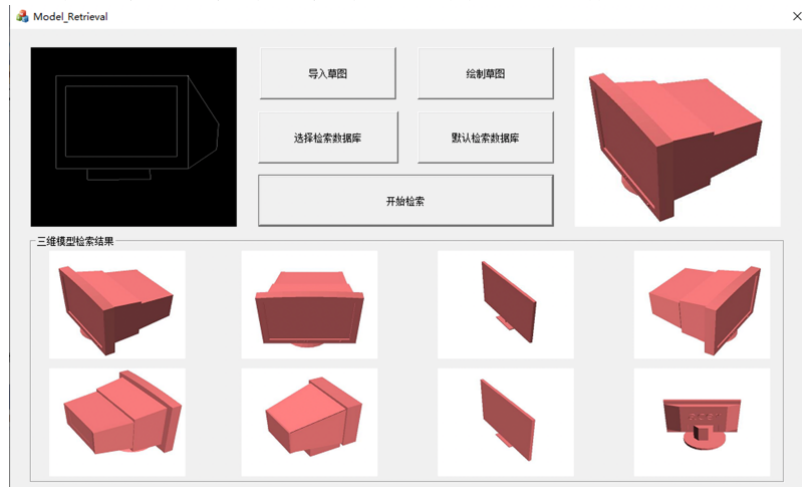


图 6-4 草图检索结果

6.3 本章小结

本章主要介绍了基于不同距离公式的相似性度量，推到了不同距离公式的计算方法和范数。本文利用欧式距离计算二维视图与草图之间的相似性，对手绘草图的检索结果进行从小到大的排序显示。本章还着重介绍了基于手绘草图的三维模型检索系统的应用。针对草图的检索过程的第一步为绘制草图，为了降低了对用户的要求，提供了在线绘制或者线下绘制的功能，只需要简单的草图即可完成检索，实验结果也表明本文的方法可以有效地检索相关的三维模型。

结论

伴随着国内外科学家们对计算机计算能力的研究，计算机的计算速度日新月异。这带来的一个显著的影响就是计算机辅助设计相关技术的蓬勃发展，三维 CAD 模型的数量从从前的零星个数到现在的成千上万。如何从海量模型数据中快速准确地检索出所需的模型变得尤为重要。如果可以快速的检索三维空间的模型，那么该技术对现有的众多产业的发展将有明显的促进作用。尤其是加工制造业，不但可以在促进新模型的更替的同时，并只需花费先前设计新模型的部分时间，节省下大量的人力与财力，减少了设计人员的工作量，缩短了产品设计周期，提高了产品设计质量。

在本文中，笔者论述了三维模型检索专业技术的全球当前研究实际情况与这个专业技术的发展进步及重要方式。在本文中，笔者主要对根据手绘简单草图的三维模型检索展开了分析研究。在根据简单草图的搜索过程里，首先利用 OpenGL 来渲染出模型，并采用固定投影的方式来获取模型点 6 张二维视图集。提供给用户不同的绘制方法，来使得用户可以线上或者线下绘制草图以此来提高识别准确率。使用了一种新的描述子，即集成描述子，该描述子有效地利用了视图的区域特征和边界轮廓特征，最后利用距离公式完成检索。实验验证本文方法效果更好。基于草图的检索算法相对于其他算法的优势在于便于普通用户检索，不需要拥有大量三维模型的细节信息就可以检索出较为准确的三维模型

本文利用 OpenGL 和 OpenCV 为几何造型平台，采用 MFC 和 ImGui 为 UI 界面，采用 C++ 为编程语言开发了一个三维模型检索。该系统集成了本文提出的所有算法，包括了简易模型的渲染，简易草图的绘制，三维模型检索。利用 ModelNet-40 完成模型库，验证了本文提出算法的优良性能和检索系统的人性化开发。

致谢

时光荏苒,岁月如梭。四年的大学生活马上就要结束了,有太多的收获和不舍。春梦秋云,聚散无常。随着离校日期的日趋渐近,毕业论文的完成也随之进入了尾声。在这短短四年的求学旅途中,不仅学习到了很多在未来工作中将要用到的知识技能,还让我学会了一些为人处世的经验,收获到了和同学间宝贵的友情。

感谢我非常尊敬的导师高雪瑶老师。本论文在高老师的悉心指导下完成的。高老师在学术研究上秉承着专注负责、精益求精的治学态度,诲人不倦的高尚师德。不仅使本人树立了远大的学习目标,还使本人明白了学多为人处事的道理。在校期间,在高老师的教导下,也逐渐做了几个项目,收获良多。本次论文从宣帝到完成,都离不开导师的良苦用心,倾注了老师大量的心血。在此,谨向高老师表达由衷的感谢。感谢大学中其他老师在学习和生活当中对我的关心和帮助。感谢于林森老师、李莎莎老师在课后听我询问问题和耐心的解答,感谢所有老师提供了一个学习氛围浓厚的校园环境,

四年前的一次选择,便一人独自跨越半个中国来到哈尔滨求学。非常荣幸能够在哈尔滨理工大学完成自己本科学习。求学生涯最想感谢的是我的父母,求学以来,已经过了16年,是他们的默默付出,鼓舞着我学习。感谢我的母亲为了让我在更好的学习环境中,走遍了家乡城市的所有学校。感谢我的父亲,在我初高中阶段,每周开车送我往返于石狮和厦门两个城市之间。在哈理工的四年中,感谢我的5个舍友们,你们一直的陪伴让我明白了友情的可贵。

感谢答辩组中的所有老师,感谢老师们在百忙之中可以抽出时间参加硕士本科生的毕业答辩,祝老师们在今后的工作和生活中健健康康,心想事成,万事如意。

参考文献

- [1] Sezgin T M, Stahovich T, Davis R. Sketch based interfaces: early processing for sketch understanding[C].The Workshop on Perceptive User Interfaces.ACM, 2001:1-8
- [2] Li B, Lu Y, Fares R. Semantic sketch-based 3D model retrieval[C]. IEEE International Conference on Multimedia and Expo Workshops. IEEE, 2013:555-558
- [3] Yuan Juefei, Abdul Rashid Hameed, Li Bo et al.A comparison of methods for 3D scene shape retrieval[J] Computer Vision and Image Understanding, 2020,201:103-120
- [4] Zhu Z, Wang X, Bai S, et al. Deep learning representation using autoencoder for 3D shape retrieval[J]. Neurocomputing, 2016,204:41-50.
- [5] Konstantinos Sfikas,Ioannis Pratikakis,Theoharis Theoharis.Ensemble of PANORAMA-based convolutional neural networks for 3D model classification and retrieval[J]Computers & Graphics.2018:208-218
- [6] 周燕,曾凡智,吴臣,罗粤,刘紫琴.基于深度学习的三维形状特征提取方法[J].计算机科学.2019(09):47-58
- [7] 张云峰.图像与三维模型匹配方法的研究及应用[D].南京:南京大学,2019.
- [8] 张艺琨,唐雁,陈强.基于多特征融合的三维模型检索[J].郑州大学学报(工学版).2019(01):1-6
- [9] 李海生,孙莉,吴晓群,蔡强,杜军平.基于模型内二面角分布直方图的非刚性三维模型检索[J].计算机辅助设计与图形学学报.2017(06): 1128-1134
- [10] 安勃卿.基于手绘草图的三维模型检索研究与实现[D].西北大学,2017.
- [11] Su H, Maji S, Kalogerakis E et al.Multi-view convolutional neural networks for 3d shape recognition[C]//Proceedings of ICCV2015, 2015
- [12] C. M. Cyr and B. B. Kimia. A similarity-based aspect-graph approach to 3D object recognition. 57(1), 2004. 3
- [13] 潘婷.基于草图的三维模型检索方法研究与应用[D].中北大学 2020
- [14] 徐旦华,辜嘉,李松毅,舒华忠.Zernike 矩的快速算法[J]东南大学学报 2002(03):32:2
- [15] Chan FHY, Lam FK.An all adder systolic structure for fast computation of moments[J].JVLSI Signal Process, 196, 12:159 175.
- [16] DALAL N, TRIGGS B. Histograms of Oriented Gradients for Human Detection[C]. Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, 2005, 1: 886-893.
- [17] 伊明.基于内容与草图的三维模型检索方法[D].哈尔滨理工大学.2020
- [18] 强会英,李雨虹,王洪申,等.基于 HMM 和仿射不变矩三维模型归类与检索算法[J].制造业自动化, 2019, 41(3): 40-46.

附录 A

英文原文

Towards 3D VR-Sketch to 3D Shape Retrieval

Abstract

Growing free online 3D shapes collections dictated re- search on 3D retrieval. Active debate has however been had on (i) what the best input modality is to trigger retrieval, and (ii) the ultimate usage scenario for such retrieval. In this paper, we offer a different perspective towards answering these questions – we study the use of 3D sketches as an input modality and advocate a VR-scenario where retrieval is conducted. Thus, the ultimate vision is that users can freely retrieve a 3D model by air-doodling in a VR environment. As a first stab at this new 3D VR-sketch to 3D shape retrieval problem, we make four contributions. First, we code a VR utility to collect 3D VR-sketches and conduct retrieval. Second, we collect the first set of 167 3D VR- sketches on two shape categories from ModelNet. Third, we propose a novel approach to generate a synthetic dataset of human-like 3D sketches of different abstract levels to train deep networks. At last, we compare the common multi-view and volumetric approaches: We show that, in contrast to 3D shape to 3D shape retrieval, volumetric point-based approaches exhibit superior performance on 3D sketch to 3D shape retrieval due to the sparse and abstract nature of 3D VR-sketches. We believe these contributions will collectively serve as enablers for future attempts at this problem.

1. Introduction

3D model retrieval has become an important topic due to a growing number of free online 3D repositories. It finds applications in CAD design, 3D printing, 3D animation and movies production, where the time required to model a 3D shape can be strongly reduced by relying on retrieval from existing 3D shape collections. Various in- put modalities have been tried – images and rough 2D sketch at first, with latest research focusing on 3D to 3D, i.e., using 3D scans or existing 3D shapes.

Despite great strides made, two salient questions still re- main (i) what constitutes the best input modality to initiate 3D retrieval, and (ii) under what usage scenario can such retrieval be best facilitated. For the former, 2D sketches and images are both in 2D, therefore can not reach the level of details desired, and the 2D-3D domain gap can also be counter-intuitive. The 3D-based paradigm on the other hand dictates existing 3D models to be readily available, which to some degree forms a “chicken-and-egg” problem (i.e., where/how to source the input model at the first place). As for the latter, apart from 2D sketches which are freely de- fined by the user, all other usage scenarios do not offer flexibility in terms of the input desired – one can not easily alter an image, not to mention a 3D

model/scan.

In this paper, we offer a new perspective on 3D shape retrieval – we advocate the use of 3D sketches as a new input modality. This new 3D-VR sketch to 3D shape paradigm not only enables detailed retrieval as per the common 3D model to 3D model setting, but also simultaneously offers a degree of flexibility found elsewhere in 2D sketch-based retrieval. Our ultimate vision is as follows: with a specific 3D model in mind, one emerges into a VR environment, roughly sketch out the model using handheld controllers, press retrieve and then relevant models would start populating the VR environment.

Our first contribution is therefore coding a VR environment for the said purpose. With this VR environment, we collected the first human VR-sketch dataset. 10 users with no artistic background were hired to produce a total of 167 3D VR-sketches from the two categories from ModelNet: chairs and bathtubs. Some examples are shown in. Since the collection process is both time- and cost-sensitive, we additionally propose the first 3D model to 3D sketch generator, and construct a synthetic dataset of 3D sketches. A key trait of our generator is that it can produce 3D sketches of different abstraction levels, effectively simulating that found in real-human sketches. Through training a series of deep 3D VR-sketch to 3D retrievals model using the real and synthetic datasets, we drive out a few important insights (i) retrieval performance drops with an increasing abstraction level, and (ii) models trained with synthetic sketches can already reach a decent performance level when tested on human sketches.

At last, we experiment with different shape/sketch representations, and state-of-the-art losses commonly used in 3D model retrieval but re-purposed for our problem. We focus on investigating the domain gap between 3D shapes and 3D sketches, due to the two key properties of sketches: sparsity: full 3D models versus sparse sketched lines, and abstraction: 3D models are geometrically perfect, while sketches are subject to deformations. In particular, we examine multi-view versus point-based 3D representations, and show that the later is more robust to both sparsity and an increased level of abstractness. We further propose an architecture with a reconstruction path to explicitly allow for change in abstractness.

In summary, our contributions include: (i) a new perspective on 3D model retrieval, where 3D VR-sketches are used for the first time to conduct retrieval, (ii) a dataset of human 3D VR-sketches, collected using a purpose-built VR environment, (iii) an approach to generate a synthetic 3D sketch with a variable level of abstractness, (iv) comprehensive evaluations using recent 3D shape retrieval models re-purposed to the task of 3D sketch-based retrieval, to drive out insights, plus a novel regularization track to address the sketch-model domain gap.

2. Related work

When dealing with 3D shape retrieval from a single image or a 3D shape, existing work is divided into two large groups based on the shape representation used: view-based or volumetric. The volumetric representation can be further

broken down to point-cloud or voxel-based. In this work we show that on the task of 3D sketch to 3D model retrieval point-cloud representation beats multi-view approaches due to better handling of the sparsity of 3D sketches.

2D image- and 3D shape-based retrieval. The retrieval problem in a multi-class setting is closely related to the shape classification problem, where the intermediate embedding of an image or a 3D shape is used for retrieval. A vanilla approach for multi-class shape classification is to use a soft max cross-entropy loss. Others works specifically tackle retrieval, where the triplet loss and its variants have become a common standard. Amongst these, represents the state-of-the-art for 3D model retrieval. It combines triplet and center losses, to solve for their respective drawbacks by using the class center as a positive sample. In this work we evaluate both the triplet and the triplet center losses on the task of 3D sketch-based retrieval and combine them with an additional Reconstruction Loss, tailored to the 3D sketch-based retrieval problem.

Due to a domain gap between images and 3D shapes, or target and query 3D models being from different distributions, Siamese or Heterogeneous network architectures can be more beneficial for a certain problem. In this work, we compare these two types of architectures for our multi-view baseline, and show that due to a strong domain gap between a 3D sketch and a 3D shape, the Heterogeneous architecture by far beats the Siamese one.

3D sketch-based 3D model retrieval. So far there was very little work on 3D sketch-based retrieval, especially in recent years. Most of them work with sketches collected using Microsoft Kinect, which not only has limited tracking accuracy, but the collection interface is also counter-intuitive, where 3D sketching is performed while visualizing 2D projections. As a result, sketches collected mostly have low fidelity and exhibit less details. Our VR-sketches are completely different: (i) visualization and sketching are both performed in 3D, and (ii) the use of the latest VR technology offers a high precision. Together, they ensure our VR-sketches are of high-fidelity and rich in details, thus more fitting for retrieval. A notable exception is the work by Giunchi et al., yet they address a different problem of 3D model retrieval for dense-color VR sketches (and optionally base 3D shapes), while we target at a more simplistic and abstract shape representation – a sparse set of single-color lines. Note that we can not find the said dataset in an open access, thus can not offer a direct comparison. Our NGVNN baseline is however already superior to the state-of-the-art used by and Non-photorealistic rendering (NPR). NPR is an old graphics and vision problem, for a detailed overview of existing methods for generating 2D NPR rendering we refer the interested reader to a recent report. The only attempt to produce the representation that resembles a 3D sketch automatically from a 3D shape was proposed by Li et al. as a concatenation of the shape views from six canonical viewpoints. They use this representation to show that the performance of their non-learning method on out-line to 3D model retrieval task significantly outperforms the performance of the 3D sketch to 3D model retrieval. This

experiment indicates that such simple shape representation is not sufficient as training data for 3D sketch-based 3D shape retrieval. Our experiments support that: The network trained on sketches with higher abstractness values significantly outperforms the network trained on clean sketches. To the best of our knowledge, we are the first to propose a method to generate abstract 3D sketches from 3D models.

3. Datasets

Collecting a full dataset of 3D human sketches is a labor intensive task. We collect a small dataset of human sketches that we use to guide the synthetic dataset generation. We as well use it as a test set to validate that the network trained on the proposed synthetic data generalizes well to human sketches. To obtain the training data, we reside to a common strategy of generating a synthetic training data.

3.1. Selected shapes

For training and testing our sketch-based retrieval models we use the repaired clean manifold meshes¹ from ModelNet10. We use all 10 classes from ModelNet10, with the total of 3958 shapes. We split the dataset into train, validation and test sets, which contain 2847, 317 and 792 shapes, respectively, ensuring a proportional split of shapes of each class between three sets.

3.2. Human 3D sketch dataset

Task. We are targeting 3D sketches created by novices, which can be viewed as an equivalent of the quick 2D sketches from the QuickDraw dataset. To enable the usage of the collected sketches for fine-grained retrieval evaluation we built a dataset of paired sketches and 3D models. We experimented with a setting, where one can observe a 3D model in VR for an unlimited duration of time, and then is asked to sketch from memory. We observed that under this scenario the participants were sometimes omitting features important to accurately testing fine-grained retrieval, and instead let the participants to sketch over a reference 3D model. We provide in the supplemental a qualitative evaluation of the sketches from memory and the retrieval performance on such inputs. To mimic the level of detail that can be expected from the 3D sketches from the imagination, we opted to use wide ribbon lines, but do not pose any constraints on sketching style or level of detail.

Participants. We selected 139 chairs and 28 bathtub shapes from ModelNet10's test set, and hired 10 participants, who have no art background or 3D sketch experience. The shapes were randomly divided into 10 subsets, consisting from 13-14 chairs and 2-3 bathtubs each. Each participant was assigned with one of these subsets. There is no duplicated shapes between subsets.

Interface. Although there are general-purpose VR painting and design software's that enable users to draw directly in 3D (such as Google's Tilt brush², and Facebook's Quill³), they do not serve our purpose in full: (i) we would like to record detailed stroke-based information, (ii) we require the option of displaying a reference model for data collection, and (iii) we want a fresh code

base for any additional functionalities in the future (e.g., sketch-based 3D editing). We implemented our custom 3D sketching environment based on Oculus Rift platform and Unity engine.

3.3 Synthetic training data generation

As a first step towards obtaining a synthetic sketch representation of a 3D shape, we extract detailed curve-networks with the method of Gori et al. This method is designed to produce a curve network that preserves well shape details, which are, though, uncommon for human novices sketches. We observe that novices not only omit small details, but tend to represent thin volumetric details with single lines and moreover often omit subset of feature lines. To obtain human sketch appearance, we focus on two aspects: level of detail and mechanical inaccuracies. We first perform details filtering and lines consolidation, followed by lines filtering. We then break the long curve chains into shorter strokes, to which we apply a set of local and global transformations.

3.3.1 Curves network extraction.

FlowRep method requires an input to be a curvature-aligned quad-dominant mesh. For processing efficiency, we simplify original triangular meshes by decimation before converting them into quad-dominant meshes using Blender, but the quality of conversion does not always comply with the requirement of FlowRep. Thus, in this stage, we filtered nearly 20% of original dataset which failed to be processed by FlowRep. This can be alleviated in the future by replying on more advanced quad-meshing algorithms.

3.3.2 Details filtering and consolidation.

The code by Gori et al. returns networks of curves in a form of chained edges of an input mesh. We first break these chains into several if the angle between two consequent edges is smaller than 135 degrees. We then remove all the chains whose length is smaller than 10% of the smallest of height and width of the bounding box of the original shape. We re-sample all the chains using Ramer-Douglas-Peucker algorithm with an accuracy parameter set to 0.02dm. Finally, we iteratively go over all the chains and compute for each chain the closest, tangentially aligned chain. If the distance between such two chains is smaller than 5% of the largest length of the two chains, the chains are substituted by their aggregate curve. These steps allow to remove small details and close to each other lines. Yet, the simplified curve networks contain many more lines than can be found in majority of 3D sketches by novices.

3.3.3 Abstraction.

In this section we describe our approach to obtain a 3D sketch with different levels of detail and mechanical inaccuracies, which we jointly refer to as a level of sketch abstractness α . We constraint α to be between 0 and 1.

Level of detail (I) To reduce the number of lines we first compute the similarity matrix using a discrete Frechet distance among chains. When computing the Frechet distance we first align the strokes at one of their end points, by translating one of the strokes. We then perform the grouping according to this

similarity matrix with Agglomerative Hierarchical Clustering, where the number of cluster $n_{cluster}$ is a function of l_a and the number of chains in the sketch n_{chains} : $n_{cluster} = \max(n_{chains} (1 - 0.8l_a)/2, 10)$. (II) We next for each cluster iteratively select a pair of two most distant lines in terms of their absolute positions and compute the mean distance from all the lines d_{mean} in the cluster to these two lines. We remove all the lines in a cluster for which the distance to any of the two selected lines is less than d_{mean} . (III) After getting the reduced set of chains, we break long chains into several shorter chains, which we refer to as strokes. We split each chain at vertices, where the curvatures are twice larger than the mean curvature of an original chain. We further filter out short strokes whose lengths are smaller than $0.2s_{max}$ to avoid tiny details, where s_{max} is the largest dimension of an input network.

Mechanical inaccuracies To mimic mechanical and perspective inaccuracies of human sketches we apply to each stroke a set of global and local deformations. We first apply a global translation, rotation and scaling, depending on the given level of abstraction l_a . To achieve this we deploy an auxiliary parameter t , which is randomly sampled from the range $[0, 1.5l_a]$. The rotation angle for each axes is then independently randomly sampled from the interval $[-10t, 10t]$ degrees. The scale factor for each axes is independently randomly sampled from $[1 - 0.1t, 1 + 0.1t]$. To obtain a translation vector we randomly sample from the surface of a sphere with its radius value randomly sampled from $[0, s_{max} t]$, where s_{max} is the largest dimension of an input network, as before. After a global stroke deformation, we apply a random translation for each stroke vertex. The translation vector is randomly sampled from the disk with a radius r_{vi} , which lies in the plane orthogonal to the stroke direction in the vertex v_i . The radius r_{vi} is randomly sampled from the range $[0, 0.1l_{stroke}]$, where $l_{stroke} \in [0, 1]$ is the length of the current stroke. We extend both ends of each stroke by p which value is randomly sampled from $[0, 0.1s_{max}]$, to reproduce human strokes appearance.

After global and local stroke deformations, we apply 3D spline interpolation, which results in smoother strokes, matching the appearance of human strokes.

For our experiments, we generate 5 synthetic datasets with 5 levels of abstraction, $[0.0, 0.25, 0.5, 0.75, 1.0]$, and obtain two mixed datasets by mixing sketches of all abstraction levels or only the sketches with 3 abstraction levels in the middle. shows example sketches obtained with different settings of l_a parameter.

4. Evaluations

We adopt the following common evaluation measures to evaluate the retrieval performance: Mean Average Precision (mAP), Normalized Discounted Cumulative Gain (NDCG), Nearest Neighbor (NN), which evaluate the ability of a model to discriminate between shape classes, and Top-k accuracy, which measure how many of the retrieval tasks have a ground-truth within top k retrieved results.

4.1. Sampling and rendering strategies

We experiment with two rendering styles for NGVNN and two sampling strategies for PointNet++.

3D sketch rendering for multi-view network. We generate 12 224×224 orthographic views of each 3D shape and 3D sketch by placing 12 virtual cameras around it every 30 degrees, as was proposed by Lee et al.. The cameras are elevated 30 degrees from the ground plane. For both 3D shapes and 3D sketches we experiment with two types of rendering styles: Phong Shading and depth maps. For 3D sketches we represent each line as a 3D tube.

Point cloud sampling. To get the point cloud representation, we first sample 10000 points from shapes and 3D sketches. For shapes we use Monte-Carlo sampling⁴ and for sketches we use equidistant sampling. We then adopt two types of sampling from the initial 10000 points to obtain a sparse set of 1024 points: random sampling or uniform sampling. The uniform sampling is obtained with the farthest point sampling. The sparse sets are obtained on-the-fly and thus might differ from one iteration to another.

4.2. Effect of sampling and rendering.

We compare different sampling methods for PointNet++ and rendering styles for NGVNN, when training with the classification loss and the triplet loss on the pairs of 3D shapes and 3D sketches rendered with λ set to 0.5. the uniform sampling for PointNet++ and depth rendering for NGVNN perform best, so we use these settings for the rest of the experiments.

It can be seen that point-based representation (dashed lines) by far outperforms the multi-view representation (solid lines) when we use the Siamese architecture for NGVNN. We thus for the rest of experiments use the heterogeneous architecture for NGVNN.

5. Conclusion

In this work we proposed the problem of 3D VR-sketch to 3D model retrieval. We first introduced a purpose-built VR environment to collect VR-sketches and conduct retrieval. We then collected a set of 3D human sketches for a subset of two shape classes from ModelNet10. We further proposed an approach for generating synthetic 3D sketches with different levels of abstraction, and demonstrated that the methods trained on our synthetic data generalize well to human sketches. Via a series of comprehensive evaluations, we find that compared to 3D shape-based retrieval, point-based shape representation is advantageous over the multi-view representation. At last, we propose an architecture with an additional sketch regularizing branch that leads to a superior performance over all the considered baselines, demonstrating the benefit of directly tackling the abstract nature of VR-sketches. We hope to have offered a valid first stab at this new problem.

附录 B

中文翻译

从虚拟三维草图到三维形状检索

摘要

越来越多的免费在线三维图形会进行二次或多次三维检索。然而，关于触发检索的最佳输入方式是什么，以及这种检索的最终使用场景，一直存在着激烈的争论。在本文中，我们提供了一个不同的角度来回答这些问题：我们研究了三维草图作为输入模态的使用，并提倡一个虚拟现实场景中进行检索。因此，最终的愿景是用户可以在虚拟现实环境中通过空中涂鸦自由检索 3D 模型。作为这个新的 3D-VR 草图对 3D 形状检索问题的第一个尝试，我们做了四个贡献。首先，我们编写了一个虚拟现实实用程序来收集三维虚拟现实草图并进行检索。其次，我们从 ModelNet 收集了第一组 167 个 3D 虚拟现实草图并把这些草图分为两类。第三，我们提出一种新的方法来产生一个不同抽象层次的人像三维草图的合成数据集来训练深层网络。最后，我们比较了常用的多视图和体绘制方法：我们发现，与三维形状到三维形状的检索相比，基于体绘制点的方法在三维草图到三维形状的检索中表现出更好的性能，这是由于三维 VR 草图的稀疏性和抽象性。我们相信，这些贡献将共同成为今后解决这一问题努力的助推器。

1. 介绍

三维模型检索已经成为一个重要的话题，由于越来越多的免费在线三维存储库。它在 CAD 设计、3D 打印、3D 动画和电影制作中都有应用，通过从现有的 3D 形状集合中检索，可以大大减少 3D 形状建模所需的时间。已经尝试了各种输入模式——即使用 3D 扫描或现有的 3D 形状，但多数还是图像和粗略的 2D 草图，目前最新的研究集中在 3D 到 3D。

尽管取得了巨大的进步，但两个突出的问题仍然存在：什么是启动 3D 检索的最佳输入模式，以及在什么使用场景下可以最好地促进此类检索。对于前者，二维草图和图像都是二维的，因此无法达到所需的细节层次，二维-三维域的差距也可能是反直观的。另一方面，基于 3D 的范例要求现有的 3D 模型随时可用，这在某种程度上形成了一个“鸡和蛋”的问题（首先在哪里/如何获取输入模型）。至于后者，除了由用户自由定义的 2D 草图外，所有其他使用场景都不能提供所需输入方面的灵活性即不能轻易改变图像，更不用说 3D 模型或者扫描了。

在本文中，我们提供了一个新的视角三维形状检索——我们提倡使用三维草图作为一种新的输入模式。这种新的 3D-VR 从草图到 3D 形状的模式不仅能够根据常见的 3D 模型到 3D 模型的设置进行详细的检索，而且同时还提供了在基于 2D 草图的检索中其他地方发现的一定程度的灵活性。

我们的最终愿景是：在考虑到一个特定的 3D 模型的情况下，一个人出现在一个虚拟现实环境中，使用手持控制器大致勾勒出模型，按住检索按钮，然后相关的模型将开始填充虚拟现实环境。

因此，我们的第一个贡献就是为上述目的编写一个虚拟现实环境。在这个虚拟现实环境中，我们收集了第一个人类虚拟现实草图数据集。10 名没有艺术背景的用户被雇佣从 ModelNet 的椅子和浴缸两个类别中制作了 167 幅 3D-VR 草图。由于收集过程对时间和成本都很敏感，因此我们还向三维草图生成器提出了第一个三维模型，并构建了一个三维草图的合成数据集。我们的生成器的一个关键特性是，它可以生成不同抽象级别的三维草图，有效地模拟真实人体草图中的三维草图。通过使用真实数据集和合成数据集对一系列深三维虚拟现实草图进行三维检索模型的训练，我们得出了一些重要的启示：随着抽象程度的提高，检索性能下降；使用合成草图训练的模型在人体草图上进行测试时，已经可以达到相当好的性能水平。

最后，我们对不同的形状/草图表示进行了实验，并对三维模型检索中常用的最新损失进行了分析，但针对我们的问题进行了重新设计。由于草图的两个关键特性：稀疏性：完整的三维模型与稀疏的草图线，以及抽象性：三维模型在几何上是完美的，而草图是受变形影响的，因此我们重点研究了三维形状与三维草图之间的域差距。特别地，我们研究了基于多视角和基于点的三维表示，并表明后者对稀疏性和抽象程度的提高都更为稳健。我们进一步提出了一个具有重构路径的架构，以明确地允许抽象性的变化。

总之，我们的贡献包括：三维模型检索的新视角，首次使用三维虚拟现实草图进行检索；使用专门构建的虚拟现实环境收集的人类三维虚拟现实草图数据集；生成具有可变抽象度的合成三维草图的方法，使用最近的三维形状检索模型进行综合评价，这些模型被重新用于基于三维草图的检索任务，以获得见解，加上一种新的正则化轨道，以解决草图模型领域的差距。

2.相关工作

当处理单个图像或三维形状的三维形状检索时，现有的工作根据使用的形状表示分为两大组：基于视图的或体积的。体积表示可以进一步细分为点云[或基于体素的。在这项工作中，我们表明，在三维草图到三维模型检索的任务中，点云表示优于多视图方法，因为它能更好地处理图像三维草图的稀疏性。

基于二维图像和三维形状的检索。多类环境下的检索问题与形状分类问题密切相关，其中图像或三维形状的中间嵌入用于检索。多类形状分类的常规方法是使用软最大交叉熵损失。其他人的工作是专门处理检索，其中三重态丢失及其变体已成为一个共同的标准。其中，引用的文献 14 代表了 3D 模型检索的最新技术。它结合了三重态和中心损耗，通过使用类中

心作为正样本来解决它们各自的缺点。在这项工作中，我们评估了基于三维草图的检索任务中的三重态和三重态中心损失，并将它们与一个额外的重建损失相结合，以适应基于三维草图的检索问题。

由于图像和三维形状之间存在域间隙，或者目标和查询三维模型来自不同的分布，暹罗或异构网络体系结构对于某个问题更为有利。在这项工作中，我们比较了这两种结构的多视图基线，并表明，由于三维草图和三维形状之间存在很大的域差距，异质结构远远优于暹罗结构。

基于三维草图的三维模型检索。到目前为止，基于三维草图的检索工作很少，尤其是近年来。他们中的大多数人使用的是使用微软 Kinect 收集的草图，这样不仅跟踪精度有限，而且收集界面也是反直观的，在可视化 2D 投影的同时进行 3D 草图绘制。因此，收集的草图大多保真度较低，显示的细节较少。我们的虚拟现实草图是完全不同的：可视化和素描都是在三维执行，和最新的虚拟现实技术的使用提供了高精度。它们共同确保了我们的虚拟现实草图具有高保真度和丰富的细节，因此更适合检索。一个值得注意的例外是 Giunchi 等人的工作，然而他们解决了一个不同的问题，即密集彩色 VR 草图（以及可选的基础 3D 形状）的 3D 模型检索，而我们的目标是更简单和抽象的形状表示——一组稀疏的单色线。需要注意的是，我们无法在开放访问中找到上述数据集，因此无法提供直接比较。然而，我们的 NGVNN[13]基线已经优于和使用的最新技术。

非真实感渲染（NPR）。NPR 是一个古老的图形和视觉问题，有关生成 2D NPR 渲染的现有方法的详细概述，请感兴趣的读者参阅最近的报告。Li 等人提出了唯一一种尝试，即产生类似于三维草图的表示，自动形成三维形状，作为六个标准视点形状视图的串联。他们用这种表示法表明，他们的非学习方法在离线到三维模型检索任务中的性能明显优于三维草图到三维模型检索的性能。实验表明，这种简单的形状表示不足以作为基于三维草图的三维形状检索的训练数据。我们的实验证明：在抽象度较高的草图上训练的网络显著优于在干净草图上训练的网络。据我们所知，我们是第一个提出从三维模型生成抽象三维草图的方法。

3. 数据集

收集完整的三维人体草图数据集是一项劳动密集型任务。我们收集了一个人体草图的小数据集，用于指导合成数据集的生成。我们也使用它作为一个测试集来验证所提出的合成数据训练的网络能够很好地推广到人体草图。为了获得训练数据，我们采用了一种生成综合训练数据的通用策略。

3.1 选择图形

为了训练和测试基于草图的检索模型，我们使用 ModelNet10 中修复的干净流形网格 1。我们使用 ModelNet10 中的所有 10 个类，总共有 3958 个形状。我们将数据集分为训练集、验证集和测试集，分别包含 2847、317 和 792 个形状，确保每个类的形状在三个集之间按比例分割。

3.2 认为草图绘制

任务。我们的目标是由新手创建的三维草图，可以将其视为来自 QuickDraw 数据集的快速二维草图的等价物。为了能够使用收集到的草图进行细粒度检索评估，我们构建了一个成对草图和三维模型的数据集。我们实验了一个场景，在这个场景中，一个人可以在虚拟现实观察一个三维模型无限长的时间，然后被要求从记忆中绘制草图。我们观察到，在这种情况下，参与者有时会忽略对精确测试细粒度检索非常重要的特征，而是让参与者在参考三维模型上绘制草图。我们在补充中提供了一个定性评价的草图从记忆和检索性能等输入。为了模仿想象中的三维草图的细节层次，我们选择使用宽的带状线，但不对草图样式或细节层次构成任何限制。

参与者。我们从 ModelNet10 的测试集中选择了 139 把椅子和 28 个浴缸形状，并雇佣了 10 名没有艺术背景或 3D 素描经验的参与者。这些形状被随机分成 10 个亚组，每个亚组由 13-14 把椅子和 2-3 个浴缸组成。每个参与者被分配了其中一个子集。子集之间没有重复的形状。

接口。尽管有一些通用的虚拟现实绘画和设计软件可以让用户直接在 3D 中绘画（比如谷歌的 Tilt brush2 和 Facebook 的 Quill3），但它们并不能完全满足我们的目的：我们希望记录基于笔划的详细信息；我们需要显示用于数据收集的参考模型的选项，以及（iii）我们希望为将来的任何附加功能（例如，基于草图的三维编辑）提供一个新的代码库。我们实现了基于 Oculus Rift 平台和 Unity 引擎的定制三维草图绘制环境。

3.3 生成综合训练数据

作为获得三维形状综合草图表示的第一步，我们使用 Gori 等人的方法提取了详细的曲线网络。这种方法的设计是为了产生一个曲线网络，它保留了良好的形状细节，这对于人类新手来说并不常见。我们观察到，新手不仅忽略了小细节，而且倾向于用单线表示薄的体积细节，而且往往忽略了特征线的子集。为了获得人体素描的外观，我们关注两个方面：细节层次和机械误差。我们首先执行细节过滤和行合并，然后执行行过滤。然后，我们将长曲线链分解为较短的笔划，并对其应用一组局部和全局变换。

3.3.1 曲线网络提取。

FlowRep 方法要求输入是曲率对齐的四边形主导网格。为了提高处理效率，在使用 Blender 将三角网格转换为四元主导网格之前，我们先对三角网格进行抽取简化，但转换质量并不总是符合 FlowRep 的要求。因此，在这个阶段中，我们过滤了几乎 20% 的原始数据集，这些数据集没有被 FlowRep 处理。这可以在未来通过更先进的四边形网格算法得到缓解。

3.3.2 过滤和合并。

Gori 等人的代码以输入网格的链边形式返回曲线网络。如果两条后继边之间的夹角小于 135 度，我们首先将这些链分成几个链。然后去除所有长度小于原始形状边界框最小高度和宽度 10% 的链。我们使用 Ramer-

Douglas-Peucker 算法对所有链重新采样，精度参数设置为 $0.02d_{\min}$ 。最后，我们迭代地遍历所有链，并为每个链计算最近的、切向对齐的链。如果这两条链之间的距离小于这两条链最大长度的 5%，则用它们的聚集曲线代替这两条链。这些步骤允许删除小细节并彼此靠近。然而，简化的曲线网络所包含的直线比新手在大多数三维草图中所能找到的要多得多。

3.3.3 抽象化

在本节中，我们将描述获得具有不同详细程度和机械不精确性的三维草图的方法，我们将其统称为草图抽象程度。我们将其限制在 0 和 1 之间。

细节层次为了减少线的数量，我们首先使用链之间的离散 Frechet 距离来计算相似矩阵。在计算 Frechet 距离时，我们首先通过平移其中一个笔划，将笔划在其端点处对齐。然后，我们使用凝聚层次聚类法根据相似矩阵进行分组，其中聚类数 n_{cluster} 是其和草图中链数 n_{chains} 的函数： $n_{\text{cluster}} = \max(n_{\text{chains}}(1 - 0.8la)/2, 10)$ 。接下来，我们根据绝对位置为每个簇迭代地选择一对最远的两条线，并计算从簇中所有线的平均到这两条线的平均距离。我们删除所有的线在一个集群中的距离，其中任何两个选定的线小于平均。在得到链的约化集之后，我们把长链分解成几个短链，我们称之为笔划。我们在顶点分割每个链，其中曲率是原始链平均曲率的两倍。我们进一步过滤掉长度小于 $0.2s_{\max}$ 的短笔划以避免微小细节，其中 s_{\max} 是输入网络的最大维数。

机械误差为了模拟人体草图的机械误差和透视误差，我们对每个草图应用一组全局和局部变形。我们首先根据给定的抽象级别应用全局平移、旋转和缩放。为了实现这一点，我们部署了一个辅助参数 t ，它是从 $[0, 1.5]$ 范围内随机采样的。然后从间隔中独立地随机取样每个轴的旋转角度 $[-10t, 10t]$ 度。每个轴的比例因子依赖于从 $[1]$ 中随机取样 -0.1 吨， $1+0.1$ 吨]。为了获得一个平移向量，我们从球面上随机采样，其半径值从 $[0, s_{\max} t]$ 中随机采样，其中 s_{\max} 是输入网络的最大维数，如前所述。在全局笔划变形之后，我们对每个笔划顶点应用随机平移。从具有半径 r_{vi} 的圆盘上随机取样平移向量，半径 r_{vi} 位于与顶点 v_i 笔划方向正交的平面上。半径 r_{vi} 从范围 $[0, 0.1la_{\text{stroke}}]$ 中随机取样，其中 $l_{\text{stroke}} \in [0, 1]$ 是当前笔划的长度。我们将每个笔划的两端延伸 p ， p 值从 $[0, 0.1s_{\max}]$ 中随机取样，以再现人类笔划的外观。

在全局和局部笔划变形后，采用三维样条插值，得到更平滑的笔划，与人体笔划的外观相匹配。

对于我们的实验，我们生成 5 个合成数据集，具有 5 个抽象级别， $[0.0, 0.25, 0.0.75, 1.0]$ ，并通过混合所有抽象级别的草图或仅在中间具有 3 个抽象水平的草图来获得两个混合数据集。图 3 c。显示了使用 la 参数的不同设置获得的示例草图。

4. 评价

4.1 采样和渲染策略

我们对 NGVNN 使用了两种渲染样式，对 PointNet++ 使用了两种采样策略。

多视图网络的三维草图绘制。我们产生 $12\ 224 \times 224$ 每个三维形状和三维草图的正交视图，每 30 度放置 12 个虚拟摄像机，如 Lee 等人提出的。摄像机被提升到离地平面 30 度的高度。对于三维形状和三维草图，我们使用两种类型的渲染样式进行实验：Phong 着色和深度贴图。对于三维草图，我们将每条线表示为一个三维管。

点云采样。为了得到点云表示，我们首先从形状和三维草图中抽取 10000 个点。对于形状，我们使用蒙特卡罗采样 4；对于草图，我们使用等距采样。然后从初始的 10000 个点开始采用两种采样方式，得到 1024 个点的稀疏集：随机采样或均匀采样。采用最远点采样法获得均匀采样。稀疏集是动态获得的，因此在不同的迭代中可能会有所不同。

4.2 采样和渲染的效果

我们比较了 PointNet++ 的不同采样方法和 NGVNN 的渲染样式，当在其设置为 0.5 渲染的 3D 形状和 3D 草图对上使用分类损失和三元组损失进行训练时。PointNet++ 的统一采样和 NGVNN 的深度渲染性能最好，因此我们在其余的实验中使用这些设置。

当我们将暹罗体系结构用于 NGVNN 时，基于点的表示（虚线）远远优于多视图表示（实线）。因此，在剩下的实验中，我们使用了 NGVNN 的异构体系结构。

5. 结论

本文提出了将三维虚拟现实草图应用于三维模型检索的问题。我们首先介绍了一个专门构建的虚拟现实环境来收集虚拟现实草图并进行检索。然后，我们从 ModelNet 收集了两个形状类的子集的一组 3D 人体草图。我们进一步提出了一种生成具有不同抽象层次的合成三维草图的方法，并证明了在我们的合成数据上训练的方法可以很好地推广到人类草图。通过一系列的综合评价，我们发现与基于三维形状的检索相比，基于点的形状表示比基于多视图的检索更具优势。最后，我们提出了一个架构，该架构带有一个额外的草图正则化分支，可以在所有考虑的基线上获得更高的性能，说明了直接处理 VR 草图的抽象本质的好处。我们希望能对这个新问题提供一个有效的初步尝试。