



山东大学
SHANDONG UNIVERSITY

毕业论文(设计)

论文(设计)题目:

基于 VR 的陶瓷表面图案示教绘制——机械臂部分

姓 名_____王宇航_____

学 号_____201900130097_____

学 院_____计算机科学与技术_____

专 业_____人工智能_____

年 级_____2019 级_____

指导教师_____赵海森_____

2023 年 5 月 25 日

摘 要

随着现代科技的不断发展，科技对于社会生产的影响越来越大，诸多传统产业都面临着在这个科技时代所出现的全新挑战，即与现代科技相结合，使用现代科技来推动产业发展。陶瓷业一直是我国的重要产业之一，而如今也遇到了这样的问题。虚拟现实技术（VR）和机械臂生产都是当前相当热门的技术，因此根据这两种技术提出了提高陶瓷生产的一种全新方式，那就是通过示教的方式，借助 VR 技术，将经验丰富的手工艺技师的陶瓷绘制技术转换为相应的路径数据，并使用机械臂在模型上完成相应的绘制任务。这是一个团队项目，本文主要讲述其机械臂部分，具体研究内容如下：

本文首先分析了当今发展陶瓷业的重要意义和当前陶瓷业所需要的相关技术发展，明确了一种全新的发展方向。其次，结合了其他人的研究，对机械臂控制和虚拟绘制的研究现状进行了一定的分析。然后，借助 D-H 表示法对机械臂进行了建模，并对机械臂的运动学进行了实现。最后，对于路径绘制进行了实现，在具体的三维模型上可以完成相应的绘制任务。

关键词：陶瓷；虚拟现实技术；机械臂；轨迹规划；三维模型绘画

ABSTRACT

With the continuous development of modern technology, the impact of technology on social production is becoming increasingly significant. Many traditional industries are facing new challenges that have emerged in this technological era, namely, combining with modern technology and using modern technology to drive industrial development. The ceramic industry has always been one of the important industries in China, and now it has also encountered such problems. Virtual reality technology (VR) and robotic arm production are currently quite popular technologies. Therefore, a new way to improve ceramic production has been proposed based on these two technologies. That is to convert the ceramic drawing technology of experienced handicraft technicians into corresponding path data through teaching and VR technology, and use a robotic arm to complete corresponding drawing tasks on the model. This is a team project, and this article mainly discusses the mechanical arm part. The specific research content is as follows:

Firstly, the importance of developing the ceramic industry today and the related technological developments required by the industry were analyzed by this article, and a new development direction was identified. Secondly, a certain analysis was conducted on the current research status of robotic arm control and virtual brush drawing, based on the research of a large number of others. Thirdly, the D-H representation is used to model the mechanical arm, and the kinematics of the mechanical arm is realized. Finally, path drawing has been implemented, and corresponding drawing tasks can be completed on specific 3D models.

Keywords: Ceramics, VR, Mechanical Arm, Trajectory Planning, 3D Model Painting

目 录

第 1 章 绪 论.....	1
1.1 选题背景和研究意义.....	1
1.1.1 任务和分工	2
1.2 研究现状.....	2
1.2.1 机械臂.....	3
1.2.2 虚拟绘制	4
1.3 论文组织结构.....	4
第 2 章 系统设计.....	5
2.1 设计需求.....	5
2.2 开发环境.....	5
2.2.1 Unity	5
2.2.2 Visual Studio	7
2.2.3 MATLAB	7
2.3 机械臂控制.....	8
2.4 模型绘制.....	10
第 3 章 系统实现.....	12
3.1 机械臂控制.....	12
3.2 Unity 中实现机械臂	16
3.3 模型绘制.....	19
第 4 章 总结与讨论.....	24
4.1 本文总结.....	24
4.2 不足和提升.....	24
4.3 发展前景.....	25
致 谢	26
参考文献	27
附录 1 文献英文原文.....	29
附录 2 文献中文译文.....	37

第1章 绪 论

1.1 选题背景和研究意义

陶瓷是我国相当出名的一种工艺美术品，最早的陶瓷可以追溯至我国的新石器时代，一路发展至今，已经成为了我国的重要产业，甚至于瓷器的英文名 china 就是因为我国在陶瓷上有着辉煌的成就才由此得名，在瓷器传入西方后，受到了欧洲贵族们热烈的追捧，逐渐成为他们日常中不可或缺的陈设，陶瓷的广泛传播加上其具有浓厚的中国特色，使得中国和瓷器紧密的结合起来了。现如今，在我国，陶瓷制品在日常用品，艺术，工业等领域都有着非常广泛的应用

但是陶瓷业发展至今，面对着智能化、科技化的新时代，也出现了许多新的需求，需要进行一定的变革，陶瓷表面图案绘制便是其中的重要部分，在传统陶瓷业中，陶瓷表面图案的绘制大多依赖于经验丰富的手工艺技师完成，如图 1-1 所示，他们需要将陶瓷一件件的完成绘制。但是，这便产生了许多问题，首先，如果要完成绘制，需要有相关的绘制经验才能更好的完成，因此，如果要扩大生产，首先要花费大量的时间和精力来教会新人，其次，人力终究是有限的，工人数量也是有限的，如果要扩大生产，这些都是很难解决的。



图 1-1 陶瓷表面图案绘制

想要解决这些问题，就离不开现在正在快速发展的机械臂技术，机械臂是当今世界上相当重要的工业器具，从 1947 年世界首台遥操作机械手诞生至今，机械

臂已经有了七十多年的发展历史，从最开始只能进行一些简单的动作到现在已经发展成了可以在各领域各行业完成复杂任务的重要器具，控制方式和工作效率都得到了极大的提升。

当然，现代机械臂的飞速发展也对陶瓷业的发展提供了相应的帮助，得益于如今机械方面的进步，如今已经有使用机械臂来代替人工来完成陶瓷绘制，从而实现大规模生产的生产方式了，但是，这种方式下的生产只是进行某种特定图案的绘制，机械臂只能完成某种特定的运动，只能在某种指定样式的陶瓷上完成，这都是目前所存在的问题，不能满足多样化，个性化的需求。

为了解决这个问题，我们想到了虚拟现实技术（VR）技术，VR 是当今世界上相当热门的技术，从 20 世纪初开始出现相关思想，整个 20 世纪都对虚拟现实技术理论进行了不断地补充和发展，到了现在的 21 世纪，已经有了许多相关的软件和系统，在许多领域如医学、军事、工业等都有着大量的应用。

因此，为了解决这两种生产方式存在的问题，结合这两种方式的优势，实现既能借助机械来实现大规模生产，又能做到人工绘制的多样化、个性化的需求，推动陶瓷业的发展，现打算采用示教的方式，基于现在的 VR 技术，把经验丰富的手工艺技师的绘制过程使用 VR 记录下来，将其路径转换为相应的机械可以读取的数据，使用机械臂来完成在陶瓷上的图案绘制任务，这样就可以实现多样且大量的生产效果。

1.1.1 任务和分工

为了实现借助 VR 来完成陶瓷表面绘制这一需求，主要任务可以分为两大部分，一部分是 VR 部分，借助 VR 来记录经验丰富的手工艺技师的绘制过程，通过示教的方式，将相关的绘制工艺路径记录下来，另一部分则是机械臂部分，将特定的路径传给机械臂之后，控制机械臂完成相应的路径并在特定的模型上完成图案绘制任务。

这两部分的完成都需要大量的工作，因此该课题被分为了两个小课题，由我和一位同学合作完成，而我则是负责其中的机械臂部分。

1.2 研究现状

目前机械臂在虚拟空间的运动能力已经可以达到比较好的精准映射，并且在

虚拟空间虚拟绘制可以给用户提供实时反馈。因此我们将机械臂部分实现分为了机械臂控制和虚拟绘制两大内容，我们主要对这两大部分进行了研究。

1.2.1 机械臂

机械臂是一个极其复杂的系统，有着参数摄动、外界干扰和未建模动态等不确定性^[1]，因此，对于不同的需求，需要对机械臂的运动进行相应的运动轨迹规划，才能完成相应的任务。

而机械臂的研究发展包括传统的刚性机械臂和较为复杂的柔性机械臂两大部分。对于传统的刚性机械臂来说，给出相应的运动学参数以及相关的控制系统来完成控制，并且这些东西一般是固定的，但是，在现实中刚性的机械臂是难以实现的，因此想要实现机械控制的精确，必须使用特殊的材料来建造机械臂，但这就意味着高昂的成本，且控制精度并不能达到特别高。而柔性机械臂则可以解决这些问题，但柔性机械臂的控制却极为复杂，需要进行大量的研究。

现在有大量研究集中在柔性机械臂上面，而柔性机械臂动力学方程的建立主要包括 Newton-Euler 公式、Lagrange 方程、变分原理、虚位移原理和 Kane 方程^[2]。

柔性机械臂的控制方式一般有刚性化处理、前馈补偿法、加速度反馈控制、被动阻尼控制、力反馈控制法、自适应控制、PID 控制、变结构控制、模糊与神经网络控制这几种^[2]。

我们此次的研究是在虚拟空间进行，主要使用的机械臂会是实现在虚拟空间的刚性机械臂，因此刚性机械臂会是我们这次研究中的主要部分。康嘉瑞等人对机器人的逆向运动学进行了初步的研究^[3]，房立金等人对于机械臂的逆运动学使用迭代算法对求解进行了改进^[4]，还有基于神经网络的逆运动学求解方式^[5]，后面为了提高求解的实时性和准确性，还提出了一种基于 ELM-SSA-SCA 的逆运动学求解方法^[6]。

在机械臂的轨迹规划方面，早就有着平面机械手的轨迹规划模型^[7]，后面还对机械臂的障碍物检测和轨迹规划进行了相应的控制约束实现^[8]，后面还使用高斯过程回归方法来对机械臂臂轨迹规划进行了研究^[9]，随着深度学习算法的大量开发，深度学习应用于机械臂轨迹规划同样近期实现的一个重要成就^[10]。

我们将借助于他们对于机械臂的运动学及轨迹规划方面的研究来实现我们的

机械臂控制研究。

1.2.2 虚拟绘制

毛笔是中国特有的一种书写和绘画工具，在中国的发展过程中有着不可替代的重要地位，而在陶瓷表面的图案绘制也通常通过毛笔来完成，因此，我们在虚拟绘制这方面，预期的实现效果是实现毛笔的虚拟绘制效果。

虚拟毛笔的建模研究主要分为两类，一种是根据经验模拟，一种是根据物理模拟^[11]。根据经验的模拟主要是通过大量的书写经验以及测试数据来实现模拟，这种实现较为简单，但是毛笔是一种极为复杂的物品，仅根据经验容易出现数据不充分，模拟效果较差的情况，而物理模拟则是直接使用构成毛笔的各种物理因素来直接组建模型，这种方式模拟效果较为准确，但计算起来极为复杂。

根据经验构建的虚拟毛笔模型目前有基于经验的参数化毛笔模型^[12]和基于压感模型的毛笔模型^[13]。根据物理模拟的虚拟毛笔模型目前有 Strassmann 的毛笔模型^[14]、Wong 的毛笔模型^[15]、基于实体造型技术的毛笔模型^[16]、Girshick 的毛笔模型^[17]、基于弹性力学的毛笔模型^[18]和 Chu 等人的毛笔模型^[19]。孙美军更是对 Chu 等人的毛笔模型进行了一定的改进^[20]，既使用了经验方法，也使用了物理方法，将两者结合，更好的实现了毛笔的模拟效果。

我们将借助他们在虚拟毛笔方面的研究来完成我们在虚拟绘制方面的研究。

1.3 论文组织结构

本文根据现代科技的发展来对陶瓷表面图案的绘制进行现代化改进，以陶瓷当代的发展背景展开，讲解了一种全新的扩大陶瓷生产的方式，设计了使用机械臂完成虚拟绘制的方式，实现了控制机械臂和在三维模型上完成绘制的任务。同时对本次研究的不足和未来进行了分析。各章节安排如下：

第一章：研究了当代陶瓷业的现状和可应用于陶瓷生产技术的发展，提出一种全新的生产方式，并对国内外有关于本课题的相关研究进行了分析。

第二章：对本课题的实现形式进行了设计和讲解，分析了如何机械臂控制和三维模型绘制，并将每个功能的实现形式进行了预估。

第三章：实现了机械臂控制和三维模型绘制的任务，得到我们需要的结果。

第四章：对当前研究存在的不足进行分析，并对于接下来的研究提出优化和

改进的方案，对该研究的未来进行展望。

第2章 系统设计

2.1 设计需求

为了实现控制机械臂按指定路径在模型上完成相应的绘制这一任务，主要需求分为两部分，一部分是要实现机械臂按照指定路径运动，另一部分是要在模型上实现相应的绘制结果。主要针对这两部分来完成相关的设计，机械臂控制实现主要分为逆运动学求解和轨迹规划两大部分，而三维模型绘制主要分为三维模型选取和模型绘制实现两大部分，系统详细实现架构如图 2-1 所示。

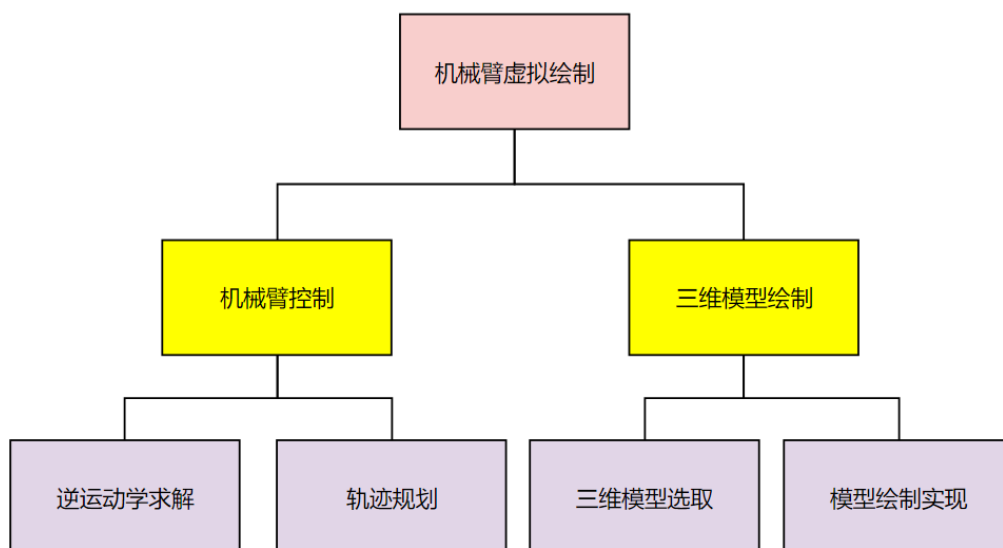


图 2-1 系统实现架构图

2.2 开发环境

2.2.1 Unity

Unity 是一个由 Unity Technologies 公司开发的实时的 3D 互动内容创作和运营平台，可以支持多种维度的开发，包括 2D、3D、VR 和 AR 等多种维度，主要的编程语言为 C#。因为其对于 VR 以及三维空间上的开发研究提供了大量的支持，我们所期望实现的是一个使用者可以做到借助 VR 来完成自己的绘制，并且这个绘制可以在相应的模型上做到实时的显示，使用者可以收到自己绘制的实时反馈，而

Unity 平台为相关的项目实现提供有大量的资源，因此将其选为了该课题的主要开发平台。该编辑器的主界面如图 2-2 Unity 工作界面图所示，由多个窗口组成。

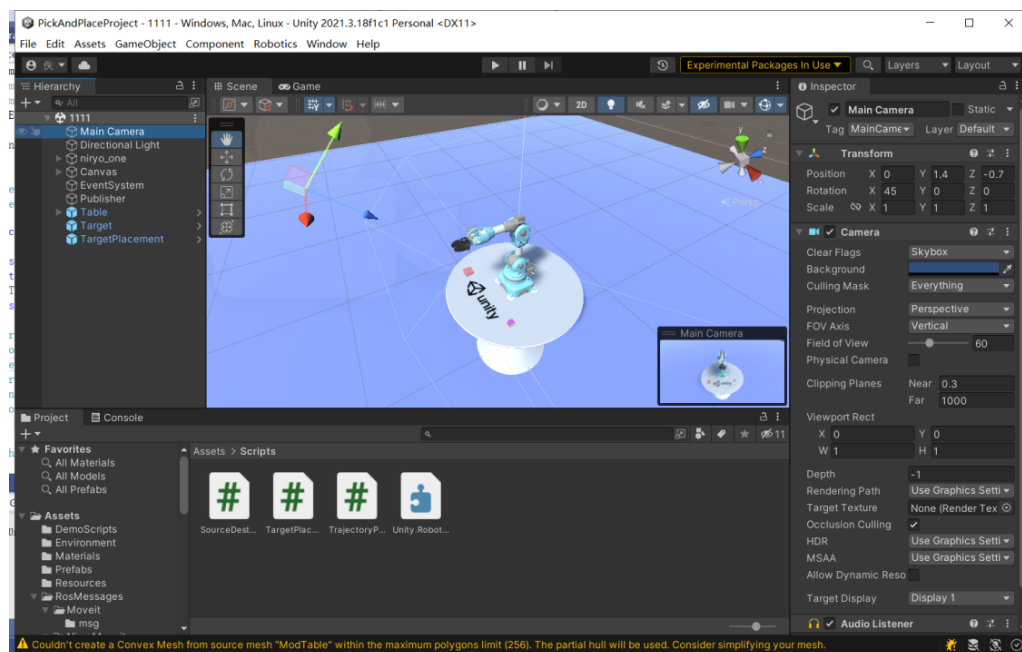


图 2-2 Unity 工作界面图

其中的 Hierarchy 窗口中包含的是该项目中的所有对象，要为一个项目中添加对象可以通过此窗口添加。

然后是 Scene 窗口，可以将项目中的所有对象进行展示，可以观察任意的项目对象。

接下来是 Game 窗口，这个窗口可以将完成的项目进行展示，表现出项目运行起来之后的样子，项目运行起来之后是什么样子，在这个界面就会展现出什么样子。

然后是 Inspector 窗口，这个窗口是对于对于选中的某个对象进行组件、数据和脚本的选取，设置该对象的详细内容。

还有就是 Project 窗口，在该窗口中放置的是这个项目中所有的资源，只要是项目中的内容都可以在这个窗口之中找到。

最后是 Console 窗口，这是项目调试的窗口，当项目在 Game 窗口进行运行时，项目运行中出现的问题都会出现在该窗口，输出的结果也会出现在该窗口。

以上就是 Unity 平台的一个简要介绍，当对各部分都有了一定的认知之后，就可以开始进行 Unity 项目的开发了。

2.2.2 Visual Studio

该平台是微软开发的一个集成开发环境，Unity 中的对象需要通过相应的脚本控制，而脚本的编写主要借助的就是这个平台来完成。主要界面如图 2-3 Visual Studio 的主要工作界面图所示，在左侧为代码编写窗口，可以在其中完成脚本的编写，右侧为项目资源窗口，可以选择不同的文件进行处理，左下方为输出窗口，输出项目中存在的错误和结果

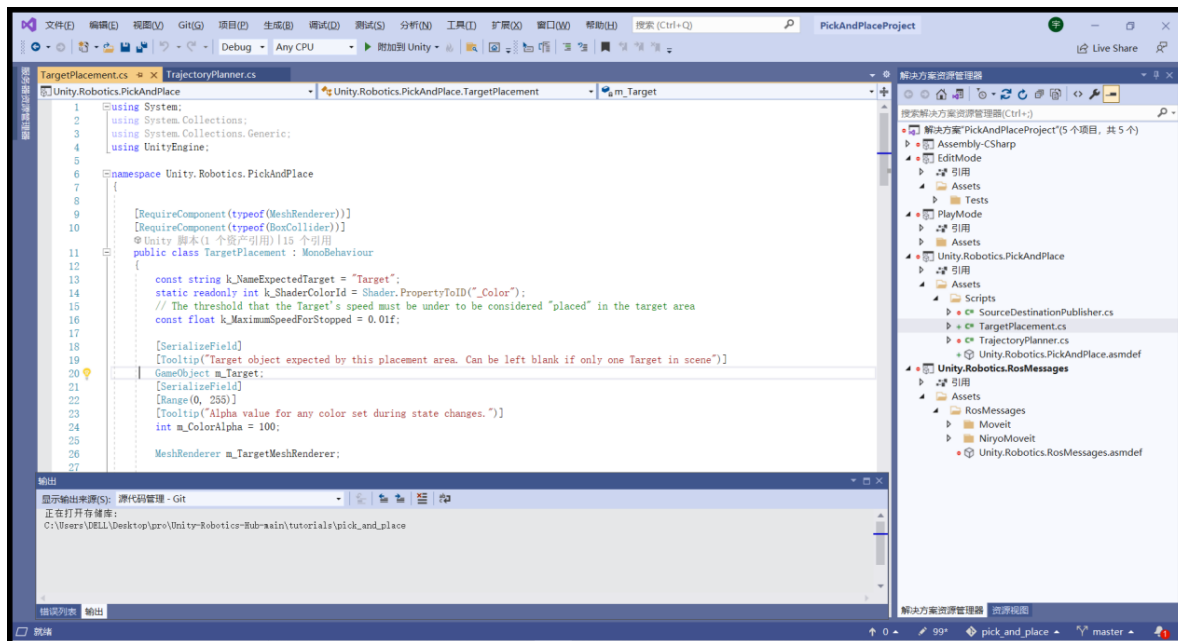


图 2-3 Visual Studio 的主要工作界面图

2.2.3 MATLAB

MATLAB 是由 MathWorks 公司开发的一款商业数学软件，在机器人等多个领域都有着非常大的作用，虽然我们的项目最终是在 Unity 中开发完成，但 Unity 在机器人领域进行的相关研究较少，且对于各种算法的实现并不能提供很好的帮助，而 MATLAB 提供的 MATLAB Robotics^[21]工具箱在机器人领域提供了大量的算法，在进行机械臂的相关研究时借助 MATLAB 便可以更加简便，因此关于机械臂控制方面的研究便基本在 MATLAB 平台完成。

其主界面如图 2-4 所示，左侧工作区中可以找到当前运行脚本中具体的变量信息，其下方则是文件选择窗口，在其中选择想要打开的文件和想进行开发的项目文件夹，右侧编辑器窗口为脚本编写区域，可以根据自己实际的需求进行脚本的编写，运行所产生的输出会在下方的命令行窗口输出，下方的命令行窗口还可以

输入一定的指令来对整个软件进行相应的处理。

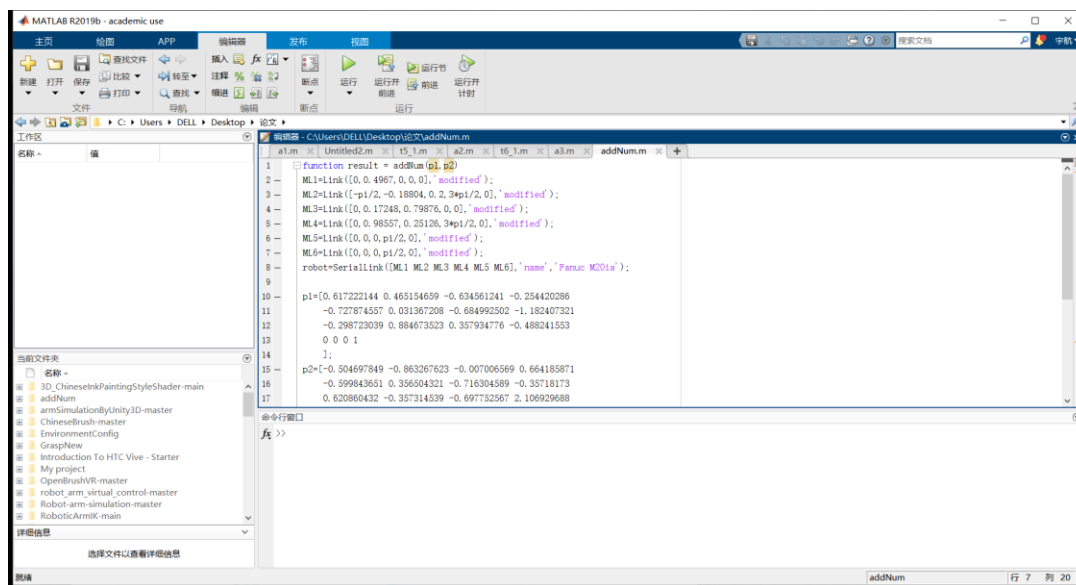


图 2-4 MATLAB 的主要工作界面

2.3 机械臂控制

要控制机械臂，就要对其正逆运动学进行研究，当所有的机械臂关节变量都知道，想求得机械臂的末端位置和姿态，这便是机械臂的正向动力学。当机械臂的末端位置和姿态已知，求各关节的变量数据时，这便是机械臂的逆向动力学。而我们要做的是根据已有路径来实现机械臂按照指定路径运动，这是在虚拟空间进行的，在研究时不对机械臂材料等问题进行研究，不考虑材料问题导致的形变等因素，因此就按照了刚性机械臂来进行了相应的研究处理，对机械臂的逆运动及其关节空间的轨迹规划进行重点研究。

早在 1955 年, Denavit 和 Hartenberg 就对机器人如何表示和建模进行了相关的研究。后面人们也都利用他们的研究来对机器人进行表示和建模，并借此来进行机器人运动学方程的推导，这也是对一个机器人进行建模的标准方法，一般称之为机器人的 D-H 表示法，建模所需的数据如连杆长度、连杆扭转、连杆偏移和关节转角等也被称之为 D-H 参数。

将连杆长度定义为 a ，连杆扭转定义为 α ，连杆偏移定义为 d ，关节转角定义为 θ ，D-H 表示法大体如图 2-5 所示，通过 D-H 表示法对机械臂进行建模，并对建模后的机械臂进行逆运动学方程求解，并完成相应的轨迹规划，从而使得机械臂按指定路径运动。

而针对于机械臂关节空间的轨迹规划，有许多不同阶次多项式轨迹规划函数以及其他类型的函数可以实现，但五次多项式的轨迹规划是相当经典的一个函数，通过位置、速度和加速度等边界条件就可以求出五次多项式的具体系数，从而使用五次多项式来完成规划轨迹的任务，这对于本课题的研究也是相当合适的，因此在关节空间进行机械臂的轨迹规划就通过该函数来完成。

最终整个机械臂控制部分的实现结构设计如图 2-6 所示。

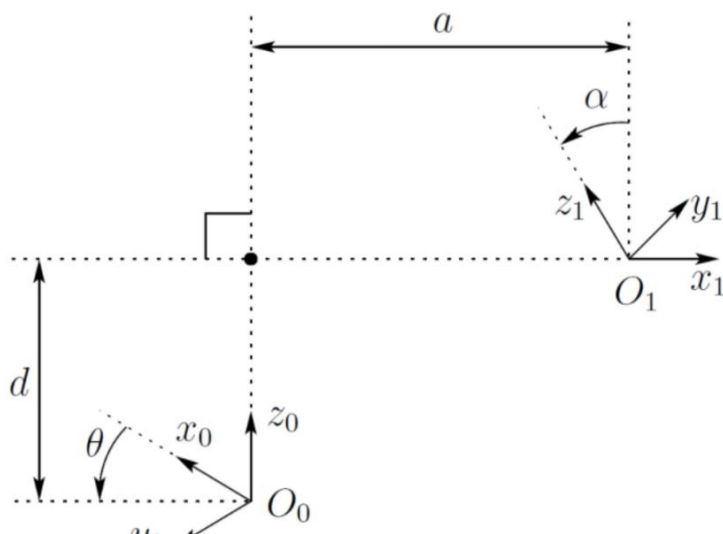


图 2-5 连杆数据的 D-H 表示

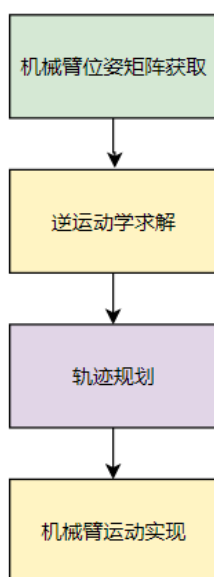


图 2-6 机械臂控制实现流程图

2.4 模型绘制

如何在模型上产生绘制结果也是一个重要部分，我们首先要明确的这个绘制是要在三维模型上来进行的，这和二维平面上的绘制是有所区别的，在二维平面上我们进行绘制，往往只需要将路径区域直接进行色素的改变就可以完成，但当其变成三维之后，这个过程便会变得更为复杂。

首先我们要做到就是进行碰撞检测，当我们的笔和这个三维模型进行接触，试图进行绘制时，这就需要一个碰撞检测算法，当碰撞实现后将相应的信息完成记录，并将想应的信息进行传递。

接下来就是对碰撞检测后记录下来的路径实现染色，将一个三维模型的具体信息直接进行更改是比较困难的，如果直接更改这个模型的颜色，只更改特定位置的话，不仅模型特定点的坐标找起来较为复杂，变色也较为麻烦，实现效果一般较差。因此，为了使染色实现更好的效果，采取了纹理渲染的方式，给模型附加一张纹理图，将模型染色转变为改变其 `RenderTexture` 的颜色来进行。

所以接下来要做的就是改变 `RenderTexture` 的颜色，我们在最开始碰撞检测时获取到了相应的路径，在检测到的路径中是有我们所需要的 UV 坐标的，我们所要做的就是将这个 UV 坐标转换为在纹理图上的具体位置，获取到了具体的位置就可以将我们的纹理图相应位置进行变色。

最后就是将新的纹理图重新于模型融合，关于融合部分，使用 Unity 中的 `ShaderGraph` 可以实现很好的融合效果，使用 `ShaderGraph` 创建一个 PBR Master（基于物理渲染主节点）即可实现模型与纹理图的融合。

整个模型绘制的大体实现流程如图 2-7 所示。

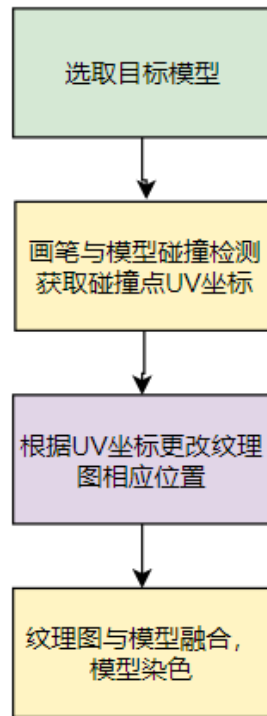


图 2-7 模型绘制实现流程图

第3章 系统实现

3.1 机械臂控制

第一步做的就是先构建一个机械臂模型，这个机械臂模型因为只是在虚拟世界进行研究，暂时还不涉及到现实应用时具体使用的机械臂，而六自由度的机械臂无论是在科学研究之中还是在日常生产之中都是比较常见的，因此在研究时就选用了比较经典的六自由度机械臂 Fanuc M20ia 进行了建模研究，使用 D-H 表示法开始建模，并且借助了 MATLAB 这个平台，参照 Fanuc M20ia 的 D-H 参数，六个连杆的数据输入如图 3-1 所示。

```
ML1=Link([0, 0.4967, 0, 0, 0], 'modified');
ML2=Link([-pi/2, -0.18804, 0.2, 3*pi/2, 0], 'modified');
ML3=Link([0, 0.17248, 0.79876, 0, 0], 'modified');
ML4=Link([0, 0.98557, 0.25126, 3*pi/2, 0], 'modified');
ML5=Link([0, 0, 0, pi/2, 0], 'modified');
ML6=Link([0, 0, 0, pi/2, 0], 'modified');
robot=SerialLink([ML1 ML2 ML3 ML4 ML5 ML6], 'name', 'Fanuc M20ia');
```

图 3-1 Fanuc M20ia 的建模函数

此时我们已经有了一个可用的机械臂模型来进行下面的研究，接下来就是对该机械臂的相关控制进行一定的研究，我们要做到的是在已知机械臂末端位姿的情况下通过机械臂的逆运动学来求出各关节变量的详细变化情况。因此首先定义了一个初始的位姿 p_1 和末端位姿 p_2 ，所要实现的就是机械臂从 p_1 位置移动到 p_2 位置这一过程。其中 p_1 、 p_2 的数据如图 3-2 所示，有了相应的位姿之后便可以据此来求得相应的关节角度的变量，分别对这两个位姿进行逆运动学求解之后，得到的初始位姿对应的变量数据为 $[-1.7699 \quad -0.0126 \quad 0.0536 \quad 0.5571 \quad 1.1697 \quad 1.2684]$ ，末端位姿对应的变量数据为 $[-0.4728 \quad -1.6417 \quad -0.6065 \quad -0.7120 \quad -1.3821 \quad -1.9695]$ 。

求得相应的关节角度变量之后，接下来便要对其进行五次多项式的轨迹规划，在这里借助了 MATLAB 平台上的机器人工具箱中的 `jtraj` 函数来实现，`itrj` 函数是 MATLAB 中机器人工具箱中用来实现五次多项式轨迹规划的函数，可以借助该函

数来计算两个关节变量之间的空间轨迹，并且能够求出速度和加速度。该函数在一定的时间内，将关节轨迹分为一定的步数来一步步实现，在研究时我选取了不同的步数进行了相应的测试，发现 40 步的时候可以取得一个较好的结果，于是就将轨迹规划的步数最终定为了 40 步。

通过函数 $[q, qd, qdd] = jtraj(init_ang, targ_ang, step)$ 完成了最终的轨迹规划，其中， q 为每一步轨迹相对应的关节角度的变量， qd 为每一步对应的速度， qdd 为每一步对应的加速度， $init_ang$ 为初始位姿对应的关节角度变量， $targ_ang$ 为末端位姿对应的关节角度变量， $step$ 为关节轨迹规划的实现步数。

实现结果如图 3-3、图 3-4、图 3-5 所示，这几张图依次为机械臂的初始位姿，机械臂的运动过程中的位姿，和机械臂的末端位姿，通过该结果可以对机械臂的运动有了一定的了解，但是并不全面。

```
p1=[0.617332144 0.465514659 -0.634651241 -0.254402286
    -0.728774557 0.031367208 -0.684992502 -1.182407321
    -0.298723039 0.884673523 0.357934776 -0.488241553
    0 0 0 1
    ];
p2=[-0.204679849 -0.863276723 -0.007006569 0.664185871
    -0.599843651 0.356502431 -0.716304589 -0.35718173
    0.620860423 -0.357341539 -0.697752567 2.106929688
    0 0 0 1
    ];
```

图 3-2 初始位姿 p1 与末端位姿 p2

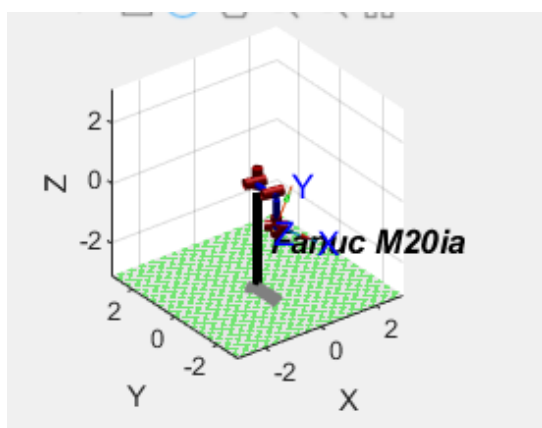


图 3-3 初始位姿

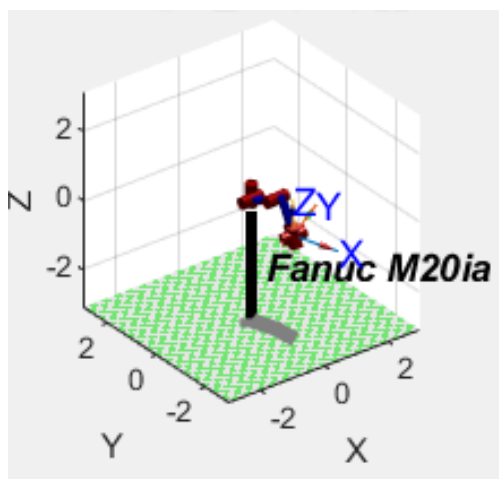


图 3-4 运动中位姿

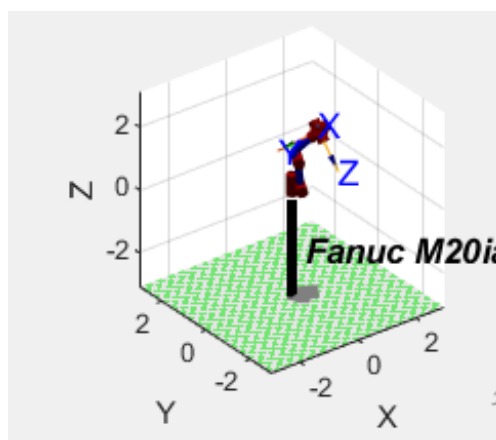


图 3-5 末端位姿

为了对机械臂运动过程进行更加深入的了解，便将各关节角度、各关节角速度、各关节角加速度、末端执行器的位置、末端执行器的速度以及末端执行器的角速度都进行了相应的研究，将它们在轨迹规划过程中的变化都借助图表的形式表示出来，借助之前求出的 q ， q_d 以及 q_{dd} 的值和 MATLAB 中的 `plot` 绘图函数便可以直接将各关节角度、各关节角速度和各关节角加速度表示出来，如图 3-6、图 3-7、图 3-8 所示，从这几张图中可以更加直观的看出在运动过程中各关节的变化过程。

机械臂的末端执行器是机械臂中相当重要的部分，机械臂的末端执行器也是最多变的一个部分，给同一个机械臂往往可以附加不同的末端执行器实现多种不同的功能，而我们后面需要完成的绘制操作也需要通过末端执行器来完成，首先进行研究的就是末端执行器的位置变化过程，在这里借助 MATLAB 机器人工具箱中的正运动学函数 `fkine`，通过已经求出的运动过程中的各关节角度变量 q 来进行正向运动学求解，求得的末端执行器的位置变化如图 3-9 所示。

接下来要求的便是末端执行器的速度和角速度变化，我们在上面已经求出了机械臂的关节速度矩阵 \dot{q} ，而机械臂的末端执行器的速度矩阵是机械臂的关节速度矩阵和雅各比矩阵的乘积，接下来就是要先得到这个雅各比矩阵，这个矩阵我们可以通过 MATLAB 机器人工具箱之中的 `jacob0` 函数和之前求出的各关节角度变量 q 来求出，求出雅各比矩阵之后我们便可以求出机械臂的一个 6×1 的速度矩阵，这个矩阵的前三个数据可以用来描述末端执行器的速度，后三个可以用来描述末端执行器的角速度，通过相应的数据便可以将速度和角速度分别表示出来，具体表示如图 3-10、图 3-11 所示。通过上面的研究我们就可以对机械臂的末端执行器有一个较为充分的研究。

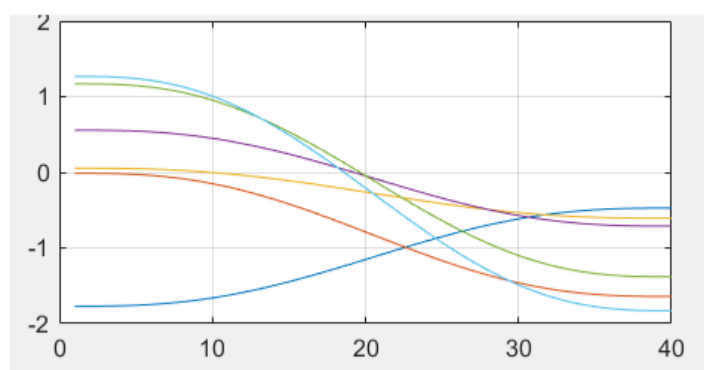


图 3-6 各关节角度随步数变化

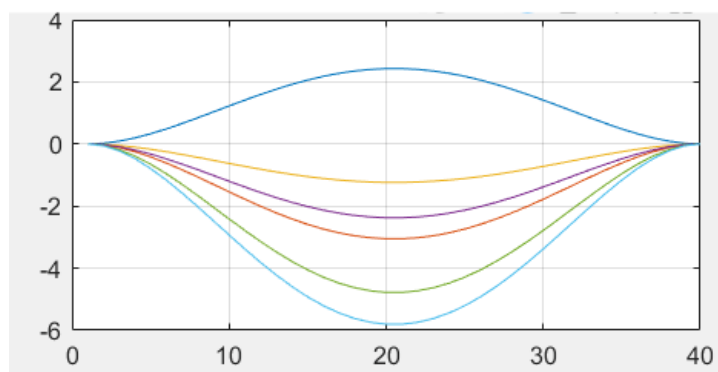


图 3-7 各关节角速度随步数变化

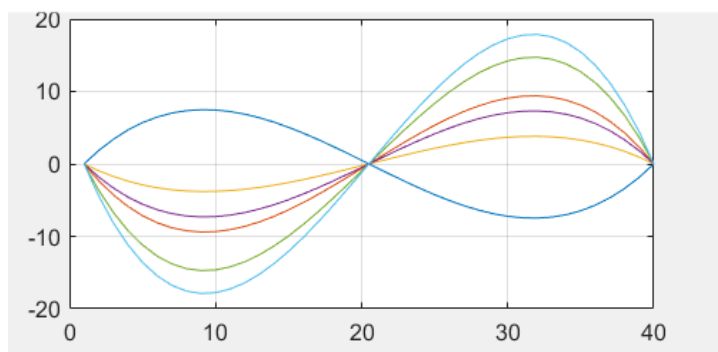


图 3-8 各关节角加速度随时间变化

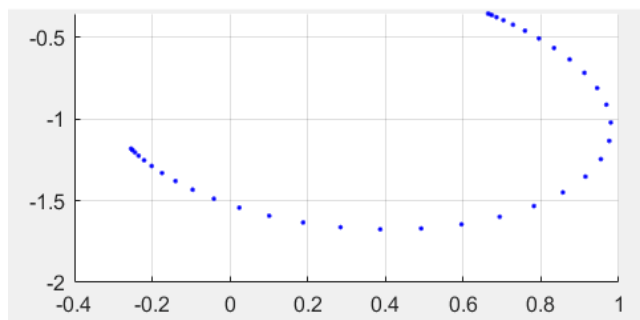


图 3-9 末端执行器的位置变化

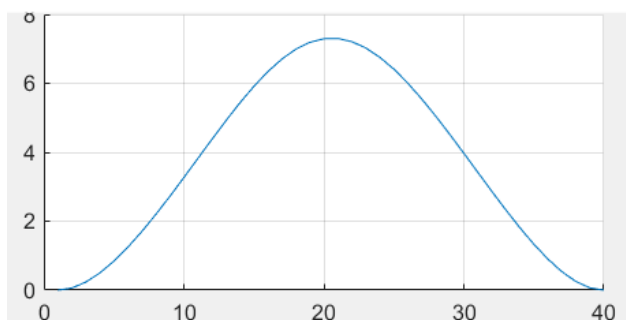


图 3-10 末端执行器的速度随步数变化

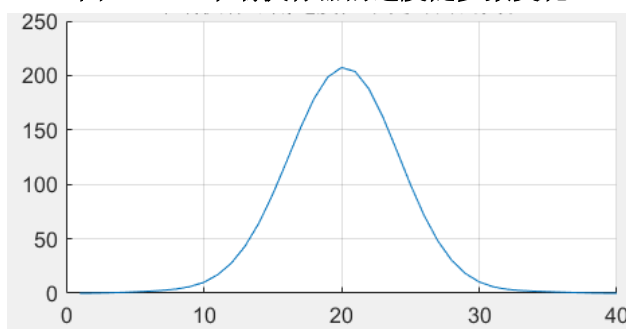


图 3-11 末端执行器的角速度随步数变化

3.2 Unity 中实现机械臂

上面我们已经借助 MATLAB 对机械臂控制进行了较为充分的研究，但是我们最终项目的实现还是要通过 Unity 这个平台来完成，因此我们要接下来要做的就是将这两个平台结合起来，实现在 Unity 中可以使用 MATLAB 的函数并完成相关研究。

实现 Unity 对 MATLAB 的调用可以有多种方式，这里采用的方式是把 MATLAB 里面的脚本文件封装为 DLL 文件，并且 Unity 完成对 DLL 的引用来完成的，要完成封装，我们首先对 DLL 进行一定的认知，DLL 是动态链接库英文 Dynamic Link Library 的缩写，使用 DLL 文件就相当于是将函数封装为了对应的应用模块，多个程序可以同时使用这一应用模块。

然后要做的就是将我们需要的函数封装为 DLL 文件，在 MATLAB 中通过 matlab compiler SDK 来完成，在命令行窗口输入 deploytool 命令之后就会出现安装的需

求,点击即可完成安装,安装完成后我们将想要调用的 MATLAB 脚本进行编写运行,得到预期结果后就可以进行封装,如图 3-12 所示,在图中为机械臂运动过程中关节角变量 q 的部分数据输出,发现我们脚本运行得到的就是预期结果,就可以通过 function 函数,将整个脚本编写为一个函数,开始封装,在命令行中输入 deploytool,完成之前的安装后便会弹出一个窗口,选择其中的 Library Compiler,或者直接在 MATLAB 的主页面上方,找到 APP 窗口也可以找到该部分,一个简要的 Library Compiler 窗口如图 3-13 所示,打开后就可以将我们之前运行好的脚本文件进行选取,并选择类型为 .NET Assembly,还有一个比较重要的细节就是对封装后的函数的命名,因为我们后续需要调用的可能会比较多,如果命名比较乱的话会使得后续的工程编写比较混乱且容易出错,在将封装的函数给出合适的命名后就开始打包,打包完成后便可以得到我们想要的 DLL 文件。

-1.7699	-0.0126	0.0536	0.5571	1.1697	1.2684
-1.7696	-0.0129	0.0535	0.5569	1.1693	1.2678
-1.7682	-0.0147	0.0528	0.5555	1.1665	1.2643
-1.7646	-0.0192	0.0510	0.5520	1.1594	1.2553
-1.7579	-0.0276	0.0475	0.5454	1.1462	1.2386
-1.7475	-0.0407	0.0422	0.5352	1.1257	1.2126
-1.7329	-0.0591	0.0348	0.5209	1.0969	1.1760
-1.7136	-0.0833	0.0250	0.5021	1.0590	1.1279
-1.6895	-0.1135	0.0127	0.4785	1.0116	1.0678
-1.6605	-0.1499	-0.0020	0.4501	0.9546	0.9955

图 3-12 关节角变量的部分数据输出

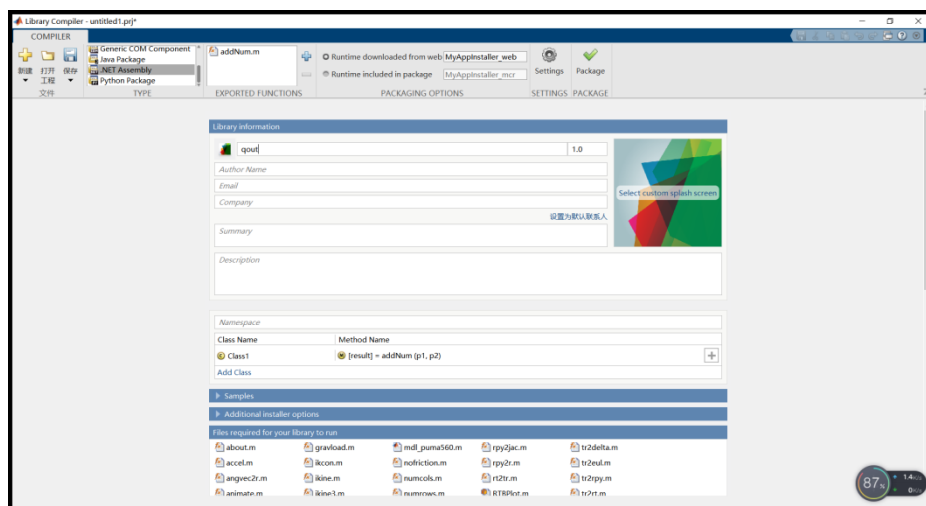


图 3-13 Library Compiler 窗口

得到我们所需要的 DLL 文件之后,接下来做的是在 Unity 添加引用,我们在 Unity 的主页面上找到 Project 窗口,打开我们项目之中的 Assets 文件夹,里面

有个名为 Plugins 的文件夹，这个文件夹负责存放我们需要添加的引用，将 DLL 文件存放于该处就可以实现引用，虽然我们现在能够调用 MATLAB 函数了，但是 MATLAB 的数据类型和 Unity 中使用的 C#数据类型有着极大的不同，数据之间并不能直接进行交换，这就需要另一个文件 MArray.dll，这是 MATLAB MathWorks 提供的一个 API，可以作为数据处理过程中的中间类，实现不同类型之间的数据转化和数据交换，将这个文件同样放于 Plugins 文件夹之中就可以实现对 MATLAB 函数的调用。

我们接下来要做的就是 Unity 中构建一个机械臂模型，在 Unity 中有很多种机械臂的搭建方式，我们采用父子联动的这种方式，每一个关节都和下一个关节构成父子关系，上一个关节进行运动时可以带动下一个关节同步进行运动，从而实现整个机械臂的运动，我们采用的是位姿矩阵输入来确定机械臂位置的方式来实现的机械臂的运动，在计算机械臂运动时是通过位姿矩阵这个 4×4 的矩阵来完成的，因此对每个关节进行了坐标系的约束，所有关节的控制都基于坐标系来进行，子关节根据父关节在坐标系变化进行相应的变化。

我们选取了较为简单的几个简单的模型来组装成一个相当简易的机械臂，或许与现实中的真实机械臂有较大差别，但仅供研究使用的话还是足够的，组装完成后初始样式如图 3-14 所示，借助这个简易的机械臂就可以开始实现我们在 Unity 中的机械臂控制了，我们通过 C# 中的 transform.parent 函数来设定相应的父子关系，并通过 transform.localRotation 函数就可以得各关节相对应的位置，将这两个关键的部分完成后，再将剩余的部分角度更改等较为容易的函数一一实现后就可以将这个机械臂进行简单的运动控制了，然后就是机械臂运动的过程展示，我们借助了 FixedUpdate 函数来实现，这个函数可以实现相隔固定的时间来反复运行，且所需的时间间隔可以在 Unity 中进行相应的设置，这样就可以将机械臂的运动过程按照固定的时间间隔来一步步的进行展示。接下来就可以展示我们的机械臂运动了，机械臂的初始状态和图 3-14 中的状态是一致的，我们将六个关节的初始角度变量设置为 0，然后将六个关节所要达到的角度依次设为 74、66、71、84、74 和 82，使用此数据进行了一次机械臂运动，机械臂在运动过程中的一个形态如图 3-15 所示，经过运动之后，机械臂的最终形态如图 3-16 所示。



图 3-14 一个简易的机械臂模型



图 3-15 机械臂运动中的状态



图 3-16 机械臂运动完成后的形态

3.3 模型绘制

我们最终是要在三维模型上完成我们绘制需求，所以先进行了一个三维模型

的选取和配置，因为我们要研究的模型最终是要应用在陶瓷绘制上的，会有多种不同的模型结构，因此选取的模型就不能选取那些简单的模型来是实现，需要研究的是较为复杂的一个模型上怎么实现想要的绘制效果。

因此，我选用了—个手部模型，手部模型的形状是相对复杂的，具有较高的研究价值，将所选的模型直接导入到了 Unity 的项目之中，导入在 Unity 中的模型—般只是—个极为简单的样式，并不会—有太多的相关属性设置，这些属性—般是要在 Unity 中完成赋予，并且在 Unity 中—般不会对导入的—个模型直接进行相应的变化操作，而是通过材质球来进行相应的改变，即将不同的材质球赋予不同的模型，使得模型表现出相应的样子，并且直接对材质球进行想要进行的改变，并使其在模型上表现出相应的改变。如手部模型—般，图 3-17 就对—个模型刚导入后的样子进行了显示，为了便于展示绘制后在模型上效果的显示，就选取了—个更容易显示绘制结果的材质球赋给了这个模型，材质球赋予完成后的得到了图 3-18 中的模型样式。



图 3-17 导入 Unity 的—个手部模型



图 3-18 赋予材质球后的手部模型

明确了我们所要绘制的模型之后，就可以开始在模型上的绘制任务了，因为我们选定了纹理绘制的这样一种绘制方式，首先就是创建一个纹理图，在 Unity 中可以新建一个 Render Texture，我们创建完成后可以对其进行相关的设置，一般而言，黑色就是相当合适的颜色，在这里我们也没有进行更改，主要更改的就是 Render Texture 的大小，选择合适的大小会使得我们最后得到的效果更加精确。在后面进行多次尝试之后，发现 2048×2048 尺寸是一个比较合适的大小，后续进行的研究也都在这个条件下进行。

接下来实现的是纹理图上实现绘制结果，由于我们的绘制是在三维实体模型上进行，为了实现三维和二维之间的转化，我们借助了 UV 坐标来实现，UV 坐标是一种应用在模型贴图上的数据坐标，这个坐标数据是处在 0-1 区间之中的，相对应的是整个贴图中的所有位置，比如说我们将 (0, 0) 坐标点对应的贴图位置为左上角，那么 (1, 1) 坐标点对应的贴图位置就是整个贴图的右下角，其余位置以此类推，这样就将整个贴图的相应位置都能通过 UV 坐标完成表示。接下来要做的就是获得我们需要的 UV 坐标，我们通过碰撞检测的方式来获取所需的坐标，当我们的画笔与三维模型发生碰撞时，便把相应的碰撞位置记录下来，并且获取到我们所需要的 UV 坐标数据，然后我们根据得到的数据转换为纹理图中的相应位置并且完成染色，就可以把我们在三维模型上的绘制结果展现在纹理图上了。

后面要解决的就是将已经记录绘制结果的纹理图和模型完成融合。这在 Unity 中实现借助了 ShaderGraph 来实现，ShaderGraph 是 Unity 中一个图形化的编程工具，可以帮助我们完成着色器的构建，通过这个工具我们创建了一个 PBR Master（基于物理渲染主节点），我们可以将许多实现的功能作为输入输入到此节点之中实现相应的融合，该节点可以输入的变量有十二个可以输入的变量，在这里我们只重点研究了其中与我们绘制实现密切相关的几个部分，图 3-19 就展示出了我们的输入部分。我们将之前的材质球的纹理、构建的新的纹理图、绘制的颜色和绘制的透明度作为输入进行了融合，实现融合后再赋给初始模型就实现了在模型上完成绘制的任务。

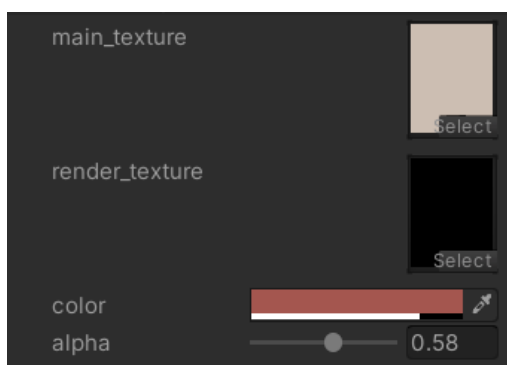


图 3-19 PBR Master 的输入

我们在模型上的绘制现在可以顺利进行了，在手部模型上进行了一次简单的绘制，绘制结果如图 3-20 所示，在不同角度对我们的绘制结果进行观察，我们可以看出在模型上绘制效果还是较好的。



图 3-20 不同角度下的模型绘制结果

我们并不满足于只实现这样简单的绘制，我们还实现了对绘制颜色进行我们想要的更改，使用了另一种颜色在我们的模型上进行绘制，取得了图 3-21 上的效果，如果说我们对于绘制的颜色深浅程度感到不满意，还可以更改我们绘制的颜色的透明度，比如我们想要绘制的颜色浅一点，就能出现图 3-22 中的效果。



图 3-21 另一种颜色下的模型绘制效果



图 3-22 更改透明度之后的绘制效果

为了使我们的研究更加全面，我们还对其它模型也进行了一定的绘制实验，我们选取了一个人体的模型，在上面实现了相应的绘制任务，并且还对笔刷的大小进行了改变，在这个模型上研究了大笔刷下的模型绘制效果。部分绘制结果在图 3-23、图 3-24 中得到展示。



图 3-23 人体模型上的绘制效果



图 3-24 更改颜色和笔刷大小后的效果

第4章 总结与讨论

4.1 本文总结

在本文中，我们根据现代科技的快速发展，尤其是机械臂相关技术以及虚拟现实技术的快速发展，为传统陶瓷业扩大生产提供了一种全新的方式，即通过私教的方式，借助 VR 技术，将经验丰富的手工艺技师的绘制经验转化为虚拟路径，使用机械臂来实现这一路径，并在模型上完成绘制。我们在本文中主要完成了机械臂部分的相关内容。

在本文中，我们探讨了如今机械臂控制的发展，分析了虚拟毛笔模型的几种构建类型，我们实现了 D-H 表示法对机械臂的建模，并对机械臂的逆运动学求解，实现了机械臂的轨迹规划，并且对机械臂在运动过程中的具体状态完成了研究。我们还做到了在 Unity 中构建一个机械臂模型，同时借助 MATLAB 函数和 Unity 函数来实现对该机械臂的控制。

我们选取了实现绘制任务的三维模型并做到了在三维模型上实现我们的绘制需求，我们还可以根据我们的需要，使用不同的颜色，并且可以选择颜色的深浅来进行绘制。

4.2 不足和提升

我们目前的研究还存在着许多的不足之处，需要后续进行相应的完善。首先就是机械臂方面的不足，我们虽然在 MATLAB 中实现了对机械臂进行相当精确的控制，但是这个控制在 Unity 之中却并没有得到良好的展现，我们在 MATLAB 中的机械臂模型和 Unity 中的机械臂模型之间存在着较大的区别，因此在机械臂控制方面的实现存在问题，现在需要找到一个合适的机械臂模型，这个模型要做到不论是在哪个平台都能完成对它的搭建和使用，从而使得机械臂能够按照我们的想法实现精确的控制。还有就是对机械臂的末端执行器进行改造，我们所需要的的是一个可以完成绘制的机械臂，这个绘制任务是要通过末端执行器来完成的，因此这个机械臂的末端执行器必须要进行改造，改造为绘制工具，并且借助该工具的数据来修改末端执行器的数据，并且使用这个机械臂新的数据进行新的建模。诸如

绘画机器人^[22]这类的便是不错的参考。

还有就是在绘制效果上的优化，目前我们在绘制方面的实现效果还存在着一一定的问题，并没有达到我们所预期实现的现实中使用毛笔所能出现的那种独特的绘制效果，接下来要做的就是想办法将我们的绘制效果向毛笔绘制那方面靠拢，或许实现一种扩散的效果会取得较好的成果，就是像毛笔那样，绘制路径中心颜色较深，向着两边，随着笔墨的逐渐扩散，颜色会随之越来越浅，这样的效果会是我们后续努力的重要方向。

后续的另一重要任务就是与 VR 部分的结合，我们已经在 Unity 上完成了 Steam VR 这个插件的安装，这是 Unity 上面进行 VR 开发的重要工具，接下来要实现的就是在 HTC Vive 这个 VR 工具与计算机实现连接，并且成功运行我们在 Unity 上的项目。

4.3 发展前景

现在市场上对于陶瓷还有着较高的需求存在的，陶瓷在艺术、工业等多个领域都有着很广泛的应用，扩大陶瓷生产的重要性是毋庸置疑的，而我们这个研究在陶瓷制造领域是有着极高的应用价值的，虽然我们是在虚拟空间中进行的相關研究，但是当我们后续的研究都一一完成后，我们就可以把虚拟空间中的成果应用到具体的现实生产当中来，我们就可以实现将本该由人力来一件件完成的陶瓷绘制任务转换为通过机械臂实现的大批量的陶瓷绘制，在我们的研究应用到具体的陶瓷制造之中后，就可以大大提高陶瓷的生产效率，不仅可以推动陶瓷产业的发展，更是能够推动对陶瓷有需求的多个领域的发展。

致 谢

本次毕业设计成功完成，首先要感谢的是赵海森教授，是他在整个过程中都一直为我的毕设提供着极大的帮助，从毕设课题的选取，到每个时期毕设进程的推进，都是他在一直支持着我，从是开题报告到中期答辩再到最后毕业论文的完成，都是他在帮助着我进行修改完善。

还要感谢的是在 Unity 方面上对我提供帮助的钟尧学长，和在机器人方面上对我提供帮助的思妍学长，正是他们在这些领域所提供的帮助才使得我的毕设可以顺利进行。

最后要感谢的是山东大学，感谢学校为我提供了一个良好的环境，使得我的毕设可以借助学校中的各种资源来顺利完成。

参考文献

- [1] 史先鹏, 刘士荣. 机械臂轨迹跟踪控制研究进展[J]. 控制工程, 2011 18(1):116-122.
- [2] 王树新, 员今天, 石菊荣, 等. 柔性机械臂建模理论与控制方法研究综述[J]. 机器人, 2002, 24(1):86-91.
- [3] 康嘉瑞, 樊留群, Roman Laurischkat. 基于 MATLAB 的机器人逆运动学研究[J]. 佳木斯大学学报(自然科学版), 2010, 28(06):824-826+848.
- [4] 房立金, 高瑞. 一般 6R 机器人逆运动学算法的改进[J]. 机械科学与技术, 2018, 37(09):1325-1330. DOI:10.13433/j.cnki.1003-8728.20180026.
- [5] Ruihua Gao. Inverse kinematics solution of Robotics based on neural network algorithms[J]. Journal of Ambient Intelligence and Humanized Computing, 2020(prepublish).
- [6] Li Guanglei, Cui Yahui, Wang Lihua, Meng Lei. Research on 6-DOF robot inverse kinematics based on blended optimization algorithm of ELM-SSA-SCA[J]. Advances in Mechanical Engineering, 2022, 14(7).
- [7] Manabu KOTANI, Takashi MIYATAKE, Haruya MATSUMOTO. A Neural Network Model for Trajectory Planning of Redundant Manipulators[J]. Transactions of the Society of Instrument and Control Engineers, 1997, 33(10).
- [8] G. PAJAK. PLANNING OF COLLISION-FREE TRAJECTORY FOR MOBILE MANIPULATORS[J]. International journal of applied mechanics and engineering, 2013, 18(2).
- [9] TRAJECTORY PLANNING OF 6-DOF MANIPULATOR BASED ON GAUSSIAN PROCESS REGRESSION METHOD[J]. International Journal of Robotics and Automation, 2020, 35(0).
- [10] Zhuang Min, Li Ge, Ding Kexin. Design and Simulation Analysis of Trajectory Planning Algorithm for 6-DoF Manipulator Based on Deep Learning[J]. Mobile Information Systems, 2022, 2022.
- [11] 侯增选, 杨广卿, 郭超, 郑栓柱. 虚拟毛笔建模研究现状与展望[J]. 计算机应用研究, 2015, 32(09):2572-2577.
- [12] 宓晓峰, 唐敏, 林建贞, 等. 基于经验的虚拟毛笔模型[J]. 计算机研究与发展, 2003, 40(8):1244-1251.
- [13] Yin Jibin, Ren Xiangshi, Ding Huaidong. HUA: an interactive calligraphy and ink-wash painting system [C] // Proc of the 5th International Conference on Computer and Information Technology. Piscataway: IEEE Press, 2005: 989-995.
- [14] Strassmann S. Hairy brushes[J]. Computer Graphics, 1986, 20(4):225-232.
- [15] Wong H T F, Ip H H S. Virtual brush: a model-based synthesis of Chinese calligraphy[J]. Computers and Graphics, 2000, 24(1):99-113.
- [16] Xu Songhua, Tang Min, Lau F, et al. A solid model based virtual hairy brush[J]. Computer Graphics Forum, 2002, 21(3):299-308.
- [17] Girshick R B. Simulation Chinese brush painting: the parametric hairy brush[C] // Proc of ACM SIGGRAPH. New York: ACM Press, 2004: 329-338.
- [18] Lee J. Physically-based modeling of brush painting[J]. Computer Networks and ISDN Systems, 1997, 29(14):1571-1576.
- [19] Chu N S H, Tai C L. Real-time painting with an expressive virtual Chinese brush[J]. IEEE

Computer Graphics and Applications,2004,24(5) : 76-85.

[20] 孙美君. 中国水墨画仿真中毛笔模型的研究与设计[D].天津: 天津大学, 2005.

[21] P.Corke.A Robotics Toolbox for MATLAB[J].IEEE Robotics and Automation Magazine,1996,3(1):24-32.

[22] 邱桂林. 绘画机器人执笔器设计与控制方法[D].沈阳工业大学,2012.

附录 1 文献英文原文

Research progress on and prospects for virtual brush modeling in digital calligraphy and painting

1 Introduction

As a tool for traditional Chinese calligraphy and painting, the brush is offered by China to the treasure house of world art. As the crown of the scholar's four jewels, the brush has occupied a cherished position in the culture of traditional Chinese painting and calligraphy since its invention. Calligraphers and painters typically express their thoughts and feelings through the help of the expressive power of brushes that can produce rich visual effects. After the beginning of the digital era, the hope was to integrate the characteristics of the brush with modern science and technology to truly simulate the function of a brush and to promote the creation of calligraphy and paintings. "A workman must sharpen his tools if he is to do his work well" (Zhu et al., 2009). To simulate the creation of calligraphy and painting naturally and vividly with the help of modern computer technology, researchers have done a great deal of work researching models for a virtual brush. They have used different modeling methods to construct models for virtual brushes that possess the characteristics of brushes. Although some achievements have been made in this respect, there are still some areas in need of development to realize the ideal goal. In this paper, we review the state of the art regarding the modeling methods of virtual brushes to provide guidance for further development.

2 State of the art and classification of virtual brush modeling research

First, we will present the state of the art on virtual brush modeling, and the main classification of the techniques. The basic principles, merits, and drawbacks of typical

modeling methods are also discussed.

2.1 Research on virtual brush modeling

Strassmann (1986) first proposed a simple two-dimensional (2D) virtual brush model, summarizing the objects involved in the process of brush painting into four elements: brush, stroke, ink dipping, and paper. This work successfully simulated Japanese “ashtray inks painting.” Subsequently, with the research and development of non-photorealistic rendering technology, various virtual brush models have been released. Chua (1990) proposed a brush model based on the outline of brush strokes, which simulates the formation of brush strokes using Bezier curves connected back and forth. Hsu and Lee (1994) proposed a brush model based on vitalization skeleton strokes; the arbitrary patterns and their evolution based on a 2D deformation model were abstracted into “ink” to express the change of brush behavior.

Lee (1997) presented a three-dimensional (3D) virtual brush model with elastic brush hair based on the theory of elasticity, which successfully painted an orchid chart with the characteristics of traditional oriental art. Lee (1999) achieved a new brush model based on the original brush model (Strassmann, 1986) by adding a geometric model of the bristle bundle and its deformation algorithm, and it can better simulate the deformation of the brush. Saito and Nakajima (1999) constructed a brush model using a Bezier curve for the skeleton of the brush head. It uses a quasi-static energy optimization method to drive the Bezier curve to simulate the morphological changes of the brush head. Baxter et al. (2001) constructed a virtual brush model with a spring particle system as the brush skeleton and subdivided the surface mesh as the brush surface. They used the deformation of the tension spring to simulate the deformation of the bristles during the painting process, and used a force feedback device for the creation of interactive art.

Chan et al. (2002) constructed a brush model consisting of many brush hairs distributed randomly in a circular area. The circular area was divided into several small squares, and the position of each brush hair in the square was random. Each

brush hair painted a line during the movement of the brush on paper, and all lines were assembled to form a stroke.

Baxter and Lin (2004) simulated the deformation of real brushes, for example, bifurcation and the plastic deformation of brushes, by improving the original brush model (Baxter et al., 2001). They used the energy minimization method to enable a novel geometric representation of the brush head. Their model can generate brushes with differently shaped brush heads, and simulate different strokes. By observing the movement of real brushes, Girshick (2004) discovered that the overall movement of the brush is closely related to the movement of a single bristle. On this basis, a new parametric brush model was constructed. The deformation of the brush was simulated by adjusting the parameters during the painting process. Adams et al. (2004) built a brush model using a point sampling surface to wrap the skeleton of the spring based on Baxter et al. (2001)'s model. The spring skeleton was used to simulate the dynamic behavior of the brush, and the point sampling surface was used to describe the change in the brush surface, and to store painting information.

Yin et al. (2005) proposed a brush model based on a pressure sensitive model. This model consists of a pressure-sensing model and a contact model, used to describe the relationship between pressure sensitivity and stroke width, and the interaction between the brush tip and the paper surface, respectively. It improved the authenticity of the user's writing with the brush, and the fluency in painting complex strokes.

van Laerhoven and van Reeth (2007) proposed a novel design method for a brush model. The geometric model of the brush constructed uses a motion chain as the spine of the brush head and a polygonal mesh as the bristle cluster. The free-form deformable grids associated with the geometric model were used to deal with various shapes of brush heads in the dynamic model. According to user input actions, an energy-optimization method was used to describe the deformation of the brush head. Baxter and Govindaraju (2010) proposed a data-driven 3D virtual brush model in which the measurement data of the actual brush deformation was stored in

tables. The deformation of the virtual brush was simulated dynamically using the data in the table. This model can effectively simulate the complex behavior of the brush, and had higher computational efficiency and good numerical stability.

We can find that the above systems either model a brush using a deformable model without individual bristles (Chu and Tai, 2002; Baxter and Lin, 2004), or model a bristle brush but ignore the simulation of 3D paint completely (Sun et al., 2009; Zhu et al., 2015). Chen et al. (2015) presented a real-time painting system that can simulate the interactions among the brush, paint, and canvas at the bristle level. They developed a variety of techniques to ensure the performance and robustness of the simulator under large time steps, including brush and particle simulations in non-inertial frames, and a fixed-point method for accelerating the Jacobi iterations. The whole system was implemented on a graphics processing unit (GPU) by the Compute Unified Device Architecture (CUDA), and overcame several challenges including the large time step, inaccuracy, and liquid transfer. However, it places a limit on the stroke speed, so artists need more time to finish paintings in their system than in the real world.

Meyer et al. (2016) proposed a tactile paintbrush to generate spatial haptic texture. Their brush can provide a method for creating stochastic friction patterns by drawing samples from a Weibull distribution for each fine-texture wavelength.

Otsuki et al. (2018) developed a mixed reality (MR) painting system, and used a visual and haptic feedback to provide the sensation of painting on virtual 3D objects using a new brush device called the “MAI Painting Brush++.” They developed a mechanism that can simulate the effect of touch and movement when the brush device is used to paint on virtual objects in the MR space, and proposed an extended model that consists of two sub-models: friction-force changing model and brush-tip spring model. They confirmed the effectiveness of the device and proposed a brush model through various user studies. However, the real-time performance of the rendering system needs to be further improved.

The research on virtual brushes in China is comparable to that in

foreign countries. Researchers from many universities and research institutes in China carried out research on virtual brushes from different perspectives in the mid and late 1990s. Yu et al. (1996) proposed a skeleton-based brush model. Based on the actual brush and actual painting process, a scatter point set was used to simulate the stroke of the brush. By changing the guiding parameters of the defined skeleton attribute function to control the distribution shape, color, and density of scatters, a more realistic painting effect can be achieved.

Ip and Wong (1997) proposed a virtual brush that uses the parametric method to synthesize how real calligraphy works. The appearance of real calligraphy can be simulated well by controlling the physical parameters related to the writing process. Wong and Ip (2000) simulated different styles of calligraphic writing and realized the design and generation of scalable fonts. Yeh et al. (2002) constructed a brush model using a bending spring as the skeleton. The brush hairs consisted of many particles connected to the bending spring and were arranged on the eight directions of the skeleton model, and the particle system was used to simulate the change of the skeleton of the brush head. Chu and Tai (2002) proposed a more accurate and effective 3D virtual brush model, by using an energy minimization method to simulate the change in the behavior of the brush, and designing a new input device, which can better simulate the different effects of the brush such as splitting of the brush head, flat head, and half-dry stroke. Xu et al. (2002) proposed a virtual brush model based on solid modeling technology, defining the primitive element as the smallest working unit of the brush model and using the aggregation of several primitives to represent the geometric shape of the brush head. The instantaneous interaction between all primitives and the paper with the cumulative effect of time forms the painting effect. Guo et al. (2002) proposed a pressure-sensitive brush model based on experimental experience and actual writing experience. Based on improvements in commercial software such as Photoshop and Elf Brush, and using a circle model as closed graphics to simulate strokes, the Bessel method was used to connect control points to form a closed curve. Then the closed curve was changed by zooming and translation to simulate the brush strokes in real time.

Mi et al. (2003) proposed an experience-based brush model, which considers only the change in the contact area between the brush and the paper surface and simulates the stroke by defining the parametric “raindrop” model driven by this basic action. In the same year, Xu et al. (2003) improved the original brush model (Xu et al., 2002) and constructed a hierarchical brush model that can express the scattering effect. Sun et al. (2005) combined the empirical model with the physical model and proposed a new 3D brush model. In the painting process, this model avoids the complexity of calculating the deformation of each bristle, greatly improving the computational efficiency and real-time interaction, and better simulating the deformation and painting effect of the brush. Bai et al. (2008) proposed a virtual brush similar to Wong and Ip (2000)’s brush model. In the process of simulating the deformation of the brush, the impression produced by the interaction between the brush and paper was used to simulate the brush stroke considering the influence of the internal and external forces acting on the brush. At the same time, according to the force acting on the brush, a set of spring meshes was used to calculate the deformation of the brush accurately. Zhu et al. (2009) proposed a virtual brush model based on statistical analysis, using the statistical analysis model to simulate the contact area between the brush and the paper surface. This model does not need to consider complicated situations such as brush modeling and brush deformation, has high computational efficiency and good real-time performance, and can achieve a better writing effect. Zhang et al. (2010) presented a virtual brush model based on triangular mesh, using a connected line segment and Bezier curve as the central axis of the brush head, and using the triangular mesh model to simulate the bristle bundle. During the writing process, the triangular mesh is driven to change by adjusting the central axis to simulatedifferent morphological changes of the brush. Zhang et al. (2014) used statistical methods to construct a parametric brushstroke model to simulate the shape of different types of pens. Combined with actual calligraphy experience, it can better generate writing animation with different styles of calligraphy.

Tang et al. (2015) modified the tactile brush algorithm to support the smooth

motion of rectangular contact areas, which can be used to convey complex touch gestures involving several simultaneous points of contact.

Yhang et al. (2015) presented a model to simulate a virtual Chinese hair brush using a camshaft curve, a novel analytic curve, to define the shape of the brush stroke. Using the camshaft curve, the shape of the stroke can be expressed flexibly via few adjustments of the parameters instead of simulation of a huge amount of brush tuft.

Li et al. (2005) proposed a calligraphy imitation system based on a virtual brush and the scale-invariant feature transform (SIFT). In their system, a virtual brush simulation algorithm was proposed. According to the rate and strength of movement of the finger on the touch screen, the trajectory of a virtual brush can be simulated, the thickness of which can be set dynamically. The system can imitate and match Chinese calligraphy based on the excellent performance of SIFT with rotation, scale, viewpoint, and illumination change, and achieves high accuracy and fast response.

Guo et al. (2017b) proposed a novel 3D interactive painting method for Chinese painting and calligraphy via real-time force feedback. The relationship between the force exerted on the brush head and the resulting brush deformation has been analyzed for the first time, and a virtual spring-mass model was applied to construct a model of the 3D brush. The 3D brush footprint can be obtained via projecting the 2D brush footprint onto the surface of the 3D virtual object in real time. The proposed methods have been applied in a virtual 3D interactive painting system via haptic feedback technology, which can effectively enhance the reality of the virtual painting process. However, their brush model does not take into account the mechanical properties of the variable stiffness that exists in a real brush.

2.2 Classification of virtual brush modeling research

Researchers have carried out various types of research on brush modeling from different perspectives, and many important results have been obtained. According to the simulation results, virtual brush modeling can be divided into empirical brush modeling and physical simulation based brush modeling.

2.2.1 Empirical brush modeling

Empirical brush modeling uses the change in the contact area between the brush and the paper surface to simulate the brush painting effect based on summing up the practical experience in creation of calligraphy and painting and much experimental experience. By adjusting the parameters obtained via experience to change the shape of the contact area, the simulation of different effects can be realized, so this approach has simple and fast calculation and good real-time performance. However, this model does not consider the characteristics of the actual brush and some other factors such as the fickle nature of brushwork, which inevitably leads to a certain discrepancy between the simulated and real effects.

2.2.2 Physical simulation based brush modeling

Physical simulation based brush modeling considers physical factors such as the shape of the brush head, elasticity, and deformation of the brush hairs to construct the physical model of a brush. On this basis, this model uses physical laws or rules to control the dynamic change in the brush during the painting process. It not only makes the creative process more realistic and intuitive, but also makes the simulation effect rival the real effect. However, the calculation process for the physical model is generally complicated, leading to inefficiency, and sometimes it cannot guarantee a real-time interaction.

附录 2 文献中文译文

数字书画中虚拟笔刷造型的研究进展与展望

1 介绍

作为中国传统书画的工具，画笔被中国提供给世界艺术的宝库。作为学者的四宝石王冠，画笔自发明以来一直在中国传统绘画和文学文化中占有宝贵的地位。书法家和画师通常通过画笔的表达力来表达他们的思想和情感，可以产生丰富的视觉效果。在数字时代开始后，希望将画笔的特点与现代科学技术相结合，真正模拟画笔的功能，促进书画的创作。一个工人必须磨利他的工具如果他想要做好他的工作（Zhu 等人，2009 年）。为了在现代计算机技术的帮助下，自然、生动地模拟书画的创作，研究人员对虚拟画笔的模型进行了大量的研究。他们使用不同的建模方法来构建具有笔刷特征的虚拟笔刷的模型。虽然在这方面已经取得了一些成就，但仍有一些领域需要发展来实现理想的目标。在本文中，我们回顾了虚拟笔刷建模方法的研究现状，为进一步的发展提供了指导。

2 虚拟刷建模研究的现状和分类

首先，我们将介绍虚拟笔刷建模的技术现状，以及技术的主要分类。并讨论了典型建模方法的基本原理、优点和缺点。

2.1 虚拟笔刷建模的研究

Strassmann（1986）首先提出了一个简单的二维（2D）虚拟笔刷模型，将笔刷绘画过程中涉及的对象总结为四个元素：笔刷、笔画、浸墨和纸张。这项工作成功地模拟了日本的“烟灰缸水墨画”。随后，随着非逼真的再生技术的研究和发展，各种虚拟笔刷模型已经发布。Chua（1990）提出了一种基于笔触轮廓的画笔模型，它使用来回连接的贝塞尔曲线来模拟笔触的形成。Hsu 和 Lee（1994）提出了基于玻璃化骨架笔画的笔刷模型；基于二维变形模型的任意模式及其演化被抽象到“墨水”中，以表示笔刷行为的变化。

Lee（1997）基于弹性理论提出了一个具有弹性刷毛的三维（3D）虚拟刷模型，

成功地绘制出了一幅具有传统东方艺术特征的兰花图。Lee（1999）通过基于原刷模型（Strassmann, 1986），添加了刷束的几何模型及其变形算法，实现了新的刷模型，可以更好地模拟刷的变形。Saito 和中岛（1999）对刷头的骨架使用贝塞尔曲线构建了一个刷头模型。采用准静态能量优化方法驱动贝塞尔曲线，模拟了刷头的形态变化。Baxter 等人（2001）构建了以弹簧粒子系统为刷面骨架的虚拟刷面模型，并将表面网格细分为刷面。他们利用张力弹簧的变形来模拟绘画过程中刷毛的变形，并使用一个力反馈装置来创作互动艺术。

Chan 等人（2002）构建了一个由随机分布的刷毛组成的刷模型。圆形区域被划分为几个小方块，每个刷毛在正方形中的位置是随机的。每根刷子在纸上移动刷子的过程中画了一条线，所有的线都被组装起来形成一个笔画。

Baxter 和 Lin（2004）通过改进原始刷模型，模拟了真实刷的变形，例如分叉和刷的塑性变形（Baxter et al., 2001）。他们使用了能量最小化的方法，使一个新的几何表示的刷头。他们的模型可以生成具有不同形状的刷子头的刷子，并模拟不同的笔画。格希克（2004）发现，通过研究真实毛刷的运动，毛刷的整体运动与单个鬃毛的运动密切相关。在此基础上，构造了一个新的参数化电刷模型。通过在涂装过程中调整参数来模拟毛刷的变形。Adams 等人（2004）基于 Baxter 等人（2001）的模型，建立了一个使用点采样表面包裹弹簧骨架的刷子模型。利用弹簧骨架模拟刷的动态行为，利用点采样面描述刷表面的变化，并存储绘画信息。

Yin 等人（2005）提出了一种基于压力敏感模型的刷子模型。该模型由压力敏感模型和接触模型组成，分别用于描述压力灵敏度和行程宽度之间的关系，以及刷尖与纸张表面之间的相互作用。它证明了用户用画笔书写的真实性，以及绘制复杂笔画的流畅性。

van 拉尔霍文和 van Reeth（2007）提出了一种新的刷模型设计方法。所构建的刷子的地理模型使用了一个运动链作为刷子头的脊柱，以及一个多边形网格作为刚毛簇。利用与几何模型相关的自由形式可变形网格来处理动态模型中各种形状的刷头。根据用户的输入动作，采用能量优化的方法来描述刷头的变形。Baxter 和 Govindaraju（2010）提出了一种数据驱动的三维虚拟刷模型，将实际刷变形的测量数据存储在表格中。利用表中的数据对虚拟刷的变形进行了命名模拟。该模型能有效地模拟电刷的复杂行为，具有较高的计算效率和良好的数值稳定性。

我们可以发现，上述系统要么使用没有单独刷毛的可变形模型建模刷（Chu 和 Tai, 2002；巴克斯特和林，2004），要么建模刚毛刷，但完全忽略 3D 绘画的模拟（Sun 等人，2009；Zhu 等人，2015）。Chen 等人（2015）提出了一个实时绘画系统，可以在刚毛水平上模拟画笔、油漆和画布之间的相互作用。他们开发了各种技术来确保模拟器在大时间步长下的性能和鲁棒性，包括在非惯性系中的刷子和粒子模拟，以及加速雅可比迭代的不动点方法。整个系统由计算统一设备架构（CUDA）在一个图形处理单元（GPU）上实现，并克服了大时间步长、不准确和液体传输等几个挑战。然而，它限制了笔画的速度，所以艺术家需要更多的时间在他们的系统中而不是在现实世界中完成绘画。

Meyer 等人（2016）提出了一种触觉画笔来产生空间触觉纹理。他们的刷子可以提供一种方法，通过从每个细纹理波长的威布尔分布中提取样本，来创建随机摩擦图案。

Otsuki 等人（2018）开发了一种混合现实（MR）绘画系统，并使用视觉和触觉反馈，使用一种名为“MAI 绘画刷++”的新刷设备，在虚拟 3D 物体上提供绘画的感觉。他们开发了一种机械技术，可以模拟使用刷子装置在 Mr 空间中绘制虚拟物体时的触摸和运动的效果，并提出了一个扩展模型，包括两个子模型：摩擦力变化模型和刷尖弹簧模型。他们证实了去除的有效性，并通过各种用户研究提出了刷模型。但是，渲染系统的实时性能还需要进一步提高。

中国对虚拟笔刷的研究与国外的研究相当。上世纪 90 年代中后期，来自中国许多大学和研究机构的研究人员从不同的角度对虚拟笔刷进行了研究。Yu 等人（1996）提出了一种基于骨架的电刷模型。根据实际笔刷和实际涂装工艺，采用散点集对笔刷的行程进行模拟。通过改变所定义的骨架属性函数的引导参数来控制散射体的分布形状、颜色和密度，可以获得更真实的绘画效果。

Ip 和 Wong（1997）提出了一种虚拟刷，使用参数方法来合成真实书法作品。通过控制与书写过程相关的物理参数，可以很好地模拟真实书法的外观。Wong 和 Ip（2000）模拟了不同风格的书法写作，实现了可伸缩字体的设计和生成。Yeh 等人（2002）使用弯曲弹簧作为骨架构建了一个刷子模型。刷毛由许多连接到弯曲弹簧上的粒子组成，排列在骨架模型的八个方向上，并利用粒子系统模拟刷头骨架量的变化。楚和泰（2002）提出了一个更准确和有效的三维虚拟刷模型，通过使用能量最小化方法来模拟变化的行为，并设计一个新的输入设备，可以更好地

模拟不同的影响刷的分裂刷头，平头，干笔飞白。Xu et al. (2002) 提出了基于实体建模技术的虚拟刷模型，将原元定义为刷模型的最小工作单元，并使用多个原元的聚合重新发送刷头的几何形状。所有原语与论文之间的内在相互作用与时间的累积效应形成了绘画效应。Guo 等人 (2002) 基于实验经验和实际写作经验提出了预先敏感的电刷模型。基于电刷和精灵刷等商业软件的改进，利用圆模型作为封闭图形模拟笔画，采用贝塞尔方法将控制点连接起来，形成闭合曲线。然后通过缩放和平移来改变闭合曲线，以实时模拟笔触。Mi 等人 (2003) 提出了一种基于经验的刷模型，该模型只考虑刷与纸面接触面积的变化，并通过定义由该基本动作驱动的参数“雨滴”模型来模拟行程。同年，Xu et al. (2003) 改进了原刷模型 (Xu et al., 2002)，构建了能够表达冲刷效应的层次刷模型。Sun 等人 (2005) 将经验模型与物理模型相结合，提出了一种新的三维电刷模型。在涂装过程中，该模型避免了计算各刚毛变形的复杂性，大大提高了计算效率和实时交互作用，且效果更好地模拟了笔刷的变形和涂装效果。Bai 等人 (2008) 提出了 Wong 和 Ip (2000) 的虚拟刷子模型。在模拟刷子变形的过程中，利用刷子与纸的相互作用产生的印象，考虑到刷子的内、外力作用的影响，来模拟刷子的行程。同时，根据作用在刷上的力，利用一组弹簧网格精确计算刷的变形。Zhu et al. (2009) 提出了基于统计分析的虚拟刷模型，利用统计分析模型模拟刷与纸张表面的接触面积。该模型不需要考虑刷建模和刷变形等复杂情况，具有较高的计算效率和良好的实时性能，可以达到更好的书写效果。Zhang 等人 (2010) 提出了基于三角形网格的虚拟刷模型，采用连接线段和贝塞尔曲线作为刷头的中心轴，利用三角形网格模型模拟刚毛束。在书写过程中，通过调整中心轴来模拟毛刷的不同形态变化，从而驱动三角形网格的变化。Zhang 等人 (2014) 使用统计学方法构建了一个参数化的笔划模型来模拟不同类型的笔的形状。结合实际的书法经验，可以更好地产生不同风格的书法创作形式。

Tang 等人 (2015) 对触觉刷算法进行了改进，以支持矩形接触区域的平滑运动，可用于传达涉及多个同时接触点的复杂触摸手势。

Yhang 等人 (2015) 提出了一种模型，使用凸轮轴曲线，一种新的分析曲线，来定义画笔笔划的形状。利用凸轮轴曲线，可以通过少量的参数修正，而不是模拟大量的刷丛，来灵活地表示行程的形状。

Li 等人 (2005) 提出了一种基于虚拟画笔和比例不变特征变换 (SIFT) 的书

法模仿系统。在该系统中，提出了一种虚拟的刷子仿真算法。根据运动的速度和强度 手指在触摸屏上，可以模拟虚拟刷的轨迹，其厚度可以动态设置。该系统能够基于良好的书法阅读能力，通过旋转、比例、视点和光照变化来模仿和匹配中国书法，并实现了较高的精度和快速的响应。

Guo 等人(2017b)提出了一种新的基于实时力反馈的中国画和三维交互绘画方法。首次分析了施加在刷头上的力与由此产生的刷头变形之间的关系，并应用虚拟弹簧质量模型构建了三维刷头模型。通过将二维笔刷足迹实时投影到三维虚拟对象的表面上，可以获得三维笔刷足迹。该方法已通过触觉反馈技术应用于虚拟三维交互式绘画系统中，可以有效地增强虚拟绘画过程的现实性。然而，他们的刷子模型并没有考虑到在真实刷子中存在的可变刚度的力学性能。

2.2 虚拟笔刷建模的分类研究

研究人员对不同领域的刷子建模进行了不同类型的研究，取得了许多重要的成果。根据仿真结果，虚拟刷建模可分为经验笔刷建模和基于物理仿真的笔刷建模。

2.2.1 经验笔刷建模

经验笔刷建模通过总结绘画的实践经验，利用刷与纸面接触面积的变化来模拟刷涂效果。通过调整实验获得的参数来改变接触面积的形状，可以实现不同效果的模拟，因此该方法计算简单快速和良好的实时性。但是，该模型没有考虑实际画笔的特点以及笔触的变化等因素，不可避免地导致模拟效果与实际效果之间存在一定的差异。

2.2.2 基于物理仿真的笔刷建模

基于物理模拟的笔刷建模考虑了笔刷头的形状、笔刷头的弹性和刷毛的变形等物理因素，以构建笔刷的物理模型。在此基础上，该模型使用物理规律或规则来控制笔刷在涂装过程中的动态变化。它不仅使创作过程更加真实和直观，而且使模拟效果能与真实效果相媲美。但是，物理模型的计算过程通常是复杂的，导致效率低下，有时不能保证实时交互。