

中图分类号:

单位代号: 10280

密 级:

学 号:

---

上海大学



硕士学位论文

---

SHANGHAI UNIVERSITY  
MASTER'S DISSERTATION

题 目	脑神经图像数据的可视分 析与交互处理
--------	-----------------------

作 者 王文斌

学科专业 软件工程

导 师 王宜敏

完成日期 2018.04.22

姓 名：王文斌

学号：16721600

论文题目：脑神经图像数据的可视分析与交互处理

## 上海大学

本论文经答辩委员会全体委员审查, 确认符合上海大学硕士学位论文质量要求。

答辩委员会签名：

主任：

委员：

导 师：

答辩日期：

姓 名：王文斌

学号：16721600

论文题目：脑神经图像数据的可视分析与交互处理

## 原创性声明

本人声明：所呈交的论文是本人在导师指导下进行的研究工作。除了文中特别加以标注和致谢的地方外，论文中不包含其他人已发表或撰写过的研究成果。参与同一工作的其他同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

签 名：\_\_\_\_\_日 期：\_\_\_\_\_

## 本论文使用授权说明

本人完全了解上海大学有关保留、使用学位论文的规定，即：学校有权保留论文及送交论文复印件，允许论文被查阅和借阅；学校可以公布论文的全部或部分内容。

（保密的论文在解密后应遵守此规定）

签 名：\_\_\_\_\_导师签名：\_\_\_\_\_日期：\_\_\_\_\_

上海大学软件工程硕士学位论文

# 脑神经图像数据的可视分析与交互 处理

姓 名：王文斌

导 师：王宜敏

学科专业：软件工程

上海大学计算机工程与科学学院

2018 年 4 月

A Dissertation Submitted to Shanghai University for the  
Degree of Master in Software Engineering

# **Visualization-assisted Analysis and Processing of Neuron Image**

MA Candidate: Wenbin Wang

Supervisor: Yimin Wang

Major: Software Engineering

**Computer Engineering and Science College, Shanghai  
University**

**April, 2018**

## 摘 要

神经元几何形态重建的研究对理解大脑的结构功能关系及信息处理方式有着极为重要的作用。本论文设计了一个基于虚拟现实的用于脑神经图像数据的可视化分析与交互处理系统。该系统能够将通过光学显微镜得到的三维神经元图像数据在虚拟现实环境下可视化并进行多种交互操作。在此基础上, 本论文设计并实现了一种虚拟现实环境下的智能交互方式, 该智能交互方式能够以半自动方式进行神经元重建。经验证, 本虚拟现实系统能够明显提升用户在进行神经元重建时的工作体验, 有效提高用户重建神经元几何形态的效率。

**关键词:** 神经元重建; 虚拟现实; Vaa3D; 可视化系统

## ABSTRACT

The study of neuron morphology reconstruction plays an important role in how the brain works. This paper designs a virtual-reality based system for visualization, analysis and interactive processing of brain nerve image data. The system can visualize three-dimensional neuron image data obtained through an optical microscope in virtual reality and perform various interactive operations. Based on this, this dissertation designs and implements an intelligent interaction method in virtual reality. This intelligent interaction method can perform neuron reconstruction in a semi-automatic manner. It has been verified that the virtual reality system can significantly enhance the user's working experience when performing neuron reconstruction, and effectively improve the efficiency of neuron reconstruction work flow.

**Keywords:** Neuron reconstruction; Virtual reality; Vaa3D; Visualization system

# 目 录

<b>第一章 绪论</b>	<b>1</b>
1.1 课题来源	1
1.2 课题研究的目的和意义	1
1.3 国内外研究概况	3
1.3.1 国外研究概况	3
1.3.2 国内研究概况	3
1.4 论文的主要研究内容	3
<b>第二章 关键技术与关键工具介绍</b>	<b>5</b>
2.1 Vaa3D 平台	5
2.1.1 Vaa3D 平台介绍	5
2.1.2 Vaa3D 平台界面与功能展示	6
2.2 虚拟现实技术	8
2.2.1 虚拟现实技术的发展	8
2.2.2 虚拟现实技术的优势	9
2.3 虚拟现实硬件设备 HTC Vive	9
<b>第三章 Vaa3D 平台中虚拟现实显示与交互的引入</b>	<b>11</b>
3.1 系统设计	11
3.1.1 系统设计目标	11
3.1.2 系统设计架构与细节	11
3.1.3 系统功能总结	15
3.2 系统实现	16
3.2.1 Vaa3D 中的基本数据结构	16
3.2.2 3D 图像数据和其他标注的显示实现	20
3.2.3 系统功能的实现	26



第四章 虚拟现实环境中的智能交互方式.....	37
4.1 智能交互方式的定义.....	37
4.1.1 神经元形态重建的三种交互方式.....	37
4.1.2 虚拟现实环境下的智能交互方式.....	38
4.2 Virtual Finger 算法.....	38
4.3 在 VR 中的改进的 Virtual Finger 算法.....	39
第五章 可视化系统工作流程与结果分析.....	42
5.1 手动与半自动重建方法的比较与分析.....	42
5.2 系统中的重建与普通重建方法的比较与分析.....	45
5.3 结论.....	47
第六章 结论与展望.....	49
6.1 结论.....	49
6.2 展望.....	49
参考文献.....	53
作者在攻读硕士学位期间公开发表的论文.....	52
作者在攻读硕士学位期间所作的项目.....	52
致 谢.....	53

# 第一章 绪论

## 1.1 课题来源

## 1.2 课题研究的目的是和意义

神经元在人类及其他哺乳类动物的大脑系统中起到了至关重要的关键作用。深入地了解大脑内部的基本连接也有可能对脑部疾病认识的突破，并为治疗开辟新的途径。神经科学的最终目标是了解神经系统的工作机制，这就需要提取神经网络中各个神经元的形态结构，进而得到真实的神经元网络的形态学结构。

神经元形态学的研究对于理解大脑的结构功能关系及信息处理方式至为关键。研究单个神经元的 3D 形态能够更精确地理解神经元的传输机制和功能，包括神经元表型，连通性，突触整合，触发性能，以及最终在神经网络中的作用<sup>[1]</sup>。因此，研究单个神经元的 3D 形态对于理解脑功能是十分重要的，一旦产生一个较优的解决方案将对神经科学产生重大影响。

多维度的显微镜图像是当前生物学研究领域不可或缺的重要数据来源。近年，随着光学显微镜技术和细胞染色技术的不断发展，得到的神经元形态数据量也随之大幅增长。然而，生物学家不得不依靠手动或半自动方法将显微镜扫描图像转换为形态学模型。鉴于神经系统中神经元的数量，这项工作势必具有极高的挑战性和复杂性，因此，一个高效、准确、便捷的重建方法对神经科学的发展至关重要。

最近，许多研究致力于开发基于光学显微镜图像的自动或半自动神经元重建算法。然而，在这领域的研究仍然是非常具有挑战性的。目前，已有的大多数算法都存在不同的缺陷，比如对噪声敏感，不适用于 GB 级图像，不适用于结构复杂图像等等，或者需要复杂且较为严格的人为干预等等。因此，希望能够通过研究，创新神经元图像数据的分析与交互方法，提出创新的神经元重建

方法,使得研究者可以更加高效、精确、便捷的从基于显微镜扫描的神经元图像数据中的到神经元形态结构的数字化表示,进而促进神经元形态学的研究。

近年来,虚拟现实技术的飞速发展和大热,吸引着无数投资人,创业者,普通消费者以及研究人员的注意力。虚拟现实是人们利用人机交互设备,把操作者与计算机生成的三维虚拟环境连结在一起。操作者通过人机交互设备,以自然的方式(如头的转动、手的运动等)向计算机送入各种动作信息,并且通过视觉、听觉以及触觉等多种感知获得三维感觉世界的技术。

虚拟现实技术为人们提供了一种全新的显示方式,从传统的 2D 屏幕显示改变为能给人身临其境感觉的 3D 显示。目前虚拟现实技术已经广泛应用在诸如游戏,电影,医疗,培训等多个方面。

特别是在医疗领域,利用虚拟现实技术的研究与应用层出不穷,例如模拟手术场景,医疗器械培训,人体器官立体可视化等等。这些研究与应用也提供了一个新的思路:如果将三维的神经元图像数据在虚拟现实环境下实现可视化,生物学家就可以在虚拟的立体三维空间中进行神经元几何形态重建的工作。在虚拟现实环境中,生物学家可以清晰地看到整个神经元图像的三维结构,能够更方便地观察图像中各个区域信号之间的关系,在三维空间中的交互操作相比传统的 2D 屏幕上的交互也更为精确,更符合人的主观意识判断。

因此本论文预想设计一套基于虚拟现实的神经元图像的可视化分析系统,即将这些 3D 神经元图像显示在虚拟现实环境中,可以更为直观地观察到整个 3D 图像的立体结构,十分方便的就能观察到 3D 图像各个角度各个维度的图像特征。本论文进一步设计了多种在虚拟现实环境中对 3D 神经元图像进行的各种交互操作,以实现一套完整的在虚拟现实环境下的神经元几何形态重建工作流程。本论文希望此系统能够应用于神经元几何形态重建工作,并且相比于传统的 2D 屏幕上的重建工作,交互操作更加方便,重建工作更加高效,重建结果更加精确。

## 1.3 国内外研究概况

### 1.3.1 国外研究概况

Hanchuan Peng, Giorgio Ascoli 等人在 2015 年提出了 BigNeuron 项目<sup>[2]</sup>。BigNeuron 是一个定义和推进单个神经元重建和分析领域先进技术的社区，并为分析 3D 神经元几何形态创建了一个通用平台。BigNeuron 的主要目标是在一个通用的开放平台上测试尽可能多的开源自动化神经元重建算法。它拥有超大规模的公开的通过多种光学显微镜技术获取的单个 3D 神经元图像数据集。算法测试结果将使用最快的超级计算机生成，然后使用计算神经科学界精心定义的多个量化标准进行比较和验证。BigNeuron 项目旨在对最先进的神经元重建算法进行客观的比较评估。

### 1.3.2 国内研究概况

国内关于利用计算机与算法进行神经元形态重建的研究相对较少，蔺想红等人在 2017 年提出了一种三位神经元几何形态的发育生成方法，采用人工基因组对基因调控网络进行编码，用基因表达的动态特性来表示神经元树突树的发育过程<sup>[6]</sup>，有助于促进神经元机制的相关研究。

## 1.4 论文的主要内容

本论文一共分为五章内容，每一章的主要内容为：

第一章 绪论，本章节首先介绍了本论文选题的背景和重要意义，其次对神经元几何形态重建的国内外研究现状作出概述；之后，简单介绍了虚拟现实技术的发展和对于本论文选题的研究意义；最后是关于本论文结构安排的介绍。

第二章首先介绍了本论文所基于的开发平台，Vaa3D 平台，以及详细介绍了本论文研究过程中使用到的关键技术与关键工具（虚拟现实技术与 HTC Vive 虚拟现实硬件设备）。

第三章首先介绍了本论文研究设计的系统的设计细节，包括系统设计目标，系统架构，系统功能模块以及系统功能概述，接着详细介绍了本论文设计的系统的关键模块的实现方法，包括神经元重建工作流程中的基本数据结构在虚拟现实环境下的显示与交互，系统中各种关键功能的实现方法等。

第四章主要是介绍了本论文所设计的一种全新的虚拟现实环境下的智能交互方式，详细介绍了该智能交互方式的实现原理，实现步骤以及优势分析。

第五章主要是对本文所做的工作内容进行总结，并展望接下来的研究方向。

## 第二章 关键技术与关键工具介绍

### 2.1 Vaa3D 平台

#### 2.1.1 Vaa3D 平台介绍

3D可视化分析工具(3D Visualization-assisted analysis), 英文简称为Vaa3D, 中文简称为挖三维, 是一个方便, 快速, 多功能的3D / 4D / 5D图像可视化和分析系统<sup>[7]</sup>。该系统台的主要目标是提供一个跨平台的开源的图形界面平台, 可以用于大规模多维度的图像数据的可视化和定量分析, 以方便和促进生物学和医学研究。Vaa3D平台支持一个非常简单但强大的插件接口, 因此可以轻松地得到拓展和增强。

Vaa3D是跨平台的, 它可以显示大型或者超大型(GB甚至TB级别)的3D图像数据和3D表面数据<sup>[7]</sup>。Vaa3D也囊括了多个功能强大的模块致力于3D图像分析(细胞分割, 神经元追踪, 脑配准, 定量测量和统计等)和数据管理。

Vaa3D 具有以下几个显著特征, 包括(i)多维度图像数据的可视化, (ii)3D图像对象的构件和定量测量, (iii)多个3D图像的比较, 融合和管理, (iv)异构图像和其相应的表面结构的可视化, 以及(v)使用其插件界面扩展更多的Vaa3D功能。

Vaa3D平台基于三种自动化流水线将多种实现图像可视化以及定量分析的复杂应用集成到自身平台中, 这三种自动流水线分别是(i)各种图像数据的聚类、分割和3D表面重建的自动化流水线, (ii)3D图像拼接融合的自动化流水线, (iii)用于神经元细胞形态重建、量化和比较分析的自动化流水线。

简而言之, Vaa3D适用于各种生物图像信息学应用, 并为开发新的高并发量的3D图像分析算法提供了一个很棒的平台, 简化了可视化分析的工作流程。

## 2. 1. 2 Vaa3D 平台界面与功能展示

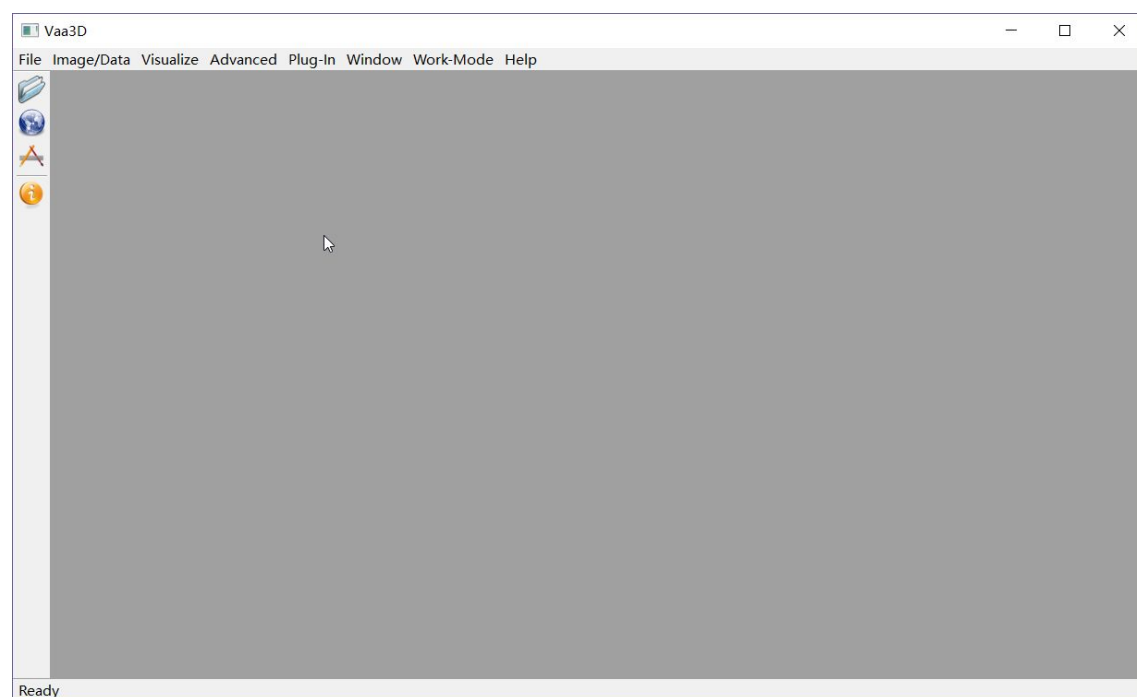


图 2. 1 Vaa3D 平台初始界面

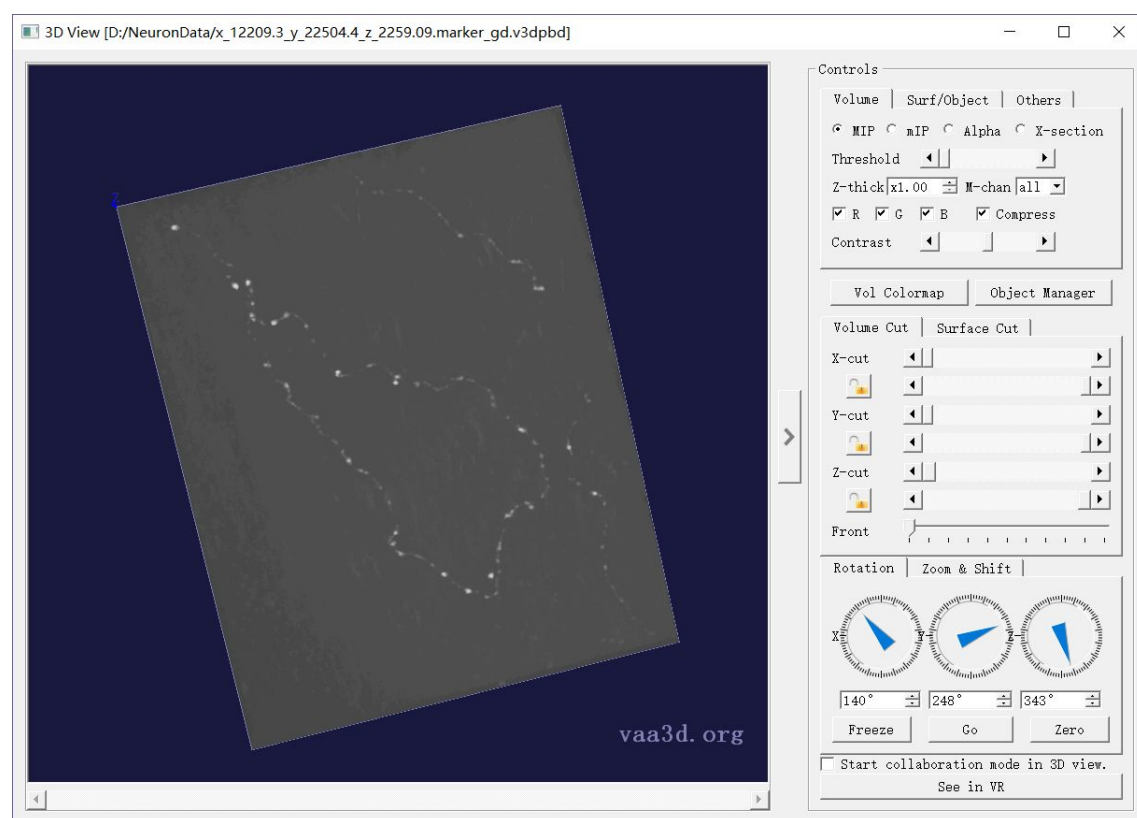


图 2.2 使用 Vaa3D 显示 3D 图像

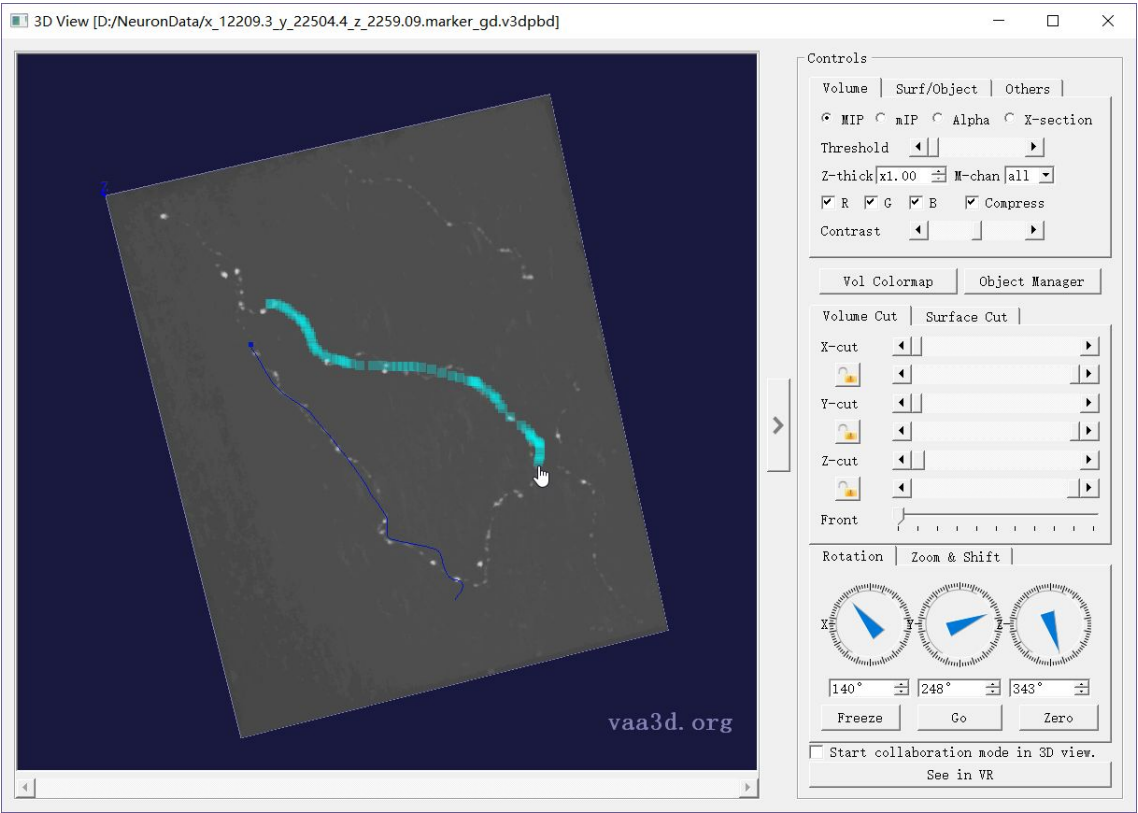


图 2.3 使用 Vaa3D 进行 neuron tracing





图 2.4 使用 Vaa3D 进行重建结果分析

## 2.2 虚拟现实技术

### 2.2.1 虚拟现实技术的发展

关于虚拟现实的定义有很多不同的说法，大致可以总结为：虚拟现实是由计算机创造的类似真实世界一样的虚拟世界，人们可以在其中像在体验真实世界一样的去体验和交互<sup>[8]</sup>。

虚拟现实概念和虚拟现实技术在经历的20世纪中期的缓慢发展阶段之后，在20世纪90年代进入大爆炸时期，期间成立了许多著名的虚拟现实产业相关的公司，比如Division, Virtuality, Fakespace<sup>[8]</sup>。各式各样的电影，书籍，会议也都开始涉及虚拟现实的研究内容。然而由于当属的技术还不够成熟，这段热潮期很快就消散了。知道2012年，Oculus VR公司成立，并随后发售了一款VR硬件产品，Oculus Rift，很快吸引了众多大型企业和媒体机构的注意，标志着虚拟现实的时代再次来临<sup>[9]</sup>。

在当前的VR竞争市场上，有三家公司牢牢占据这业界“三巨头”的位置，分

别是开发Oculus Rift的Oculus VR公司（现已经被Facebook收购），联合开发HTC Vive的HTC与Valve公司，以及开发PSVR的索尼公司。这三家公司的产品各具优势，紧紧把握着高端VR市场。

虚拟现实应用现在已经涵盖了我们生活的方方面面，包括：

① 培训教育。医疗手术培训，汽车维修培训，火车驾驶培训等危险性较高或成本较高的培训可以使用虚拟现实技术来规避这些缺点。

② 运动健身。辅以虚拟现实技术的运动可以让使用者足不出户体验到多种“真实”的运动体验。

③ 游戏娱乐。VR游戏应用占到了当前所有VR应用的60% 以上。辅以虚拟现实技术的游戏能够给玩家带来更完美的体验。

④ 广告、电影、直播等娱乐活动。

### 2.2.2 虚拟现实技术的优势

虚拟现实技术为人们获取信息方式提供了一次极大的更新，不同于移动互联网和手机让人们可以随时随地地获取外界信息，而是让人们更进一步身历其境地获取信息。随着越来越多的创业者和开发者进入虚拟现实领域，相信虚拟现实技术会越来越成熟，越来越亲近消费者。

## 2.3 虚拟现实硬件设备 HTC Vive

HTC Vive 是由 HTC 公司和 Valve 公司合作在 2016 年推出的一款虚拟现实硬件显示设备，它搭配两个无线手柄控制器，并具备手柄追踪功能，如图 2.5 所示。HTC Vive 的主要特色是能进行较大范围的移动，而且具有低延时，高精度的特点<sup>[10]</sup>。



图 2.5 HTC Vive 设备

HTC Vive 的定位原理是光塔以极短的固定间隔向外发射红外激光，由头显于手柄上的多个光感应器接收红外激光并计时，最终计算出各个设备的位置和朝向<sup>[11]</sup>。HTC Vive 可以实现一对光塔同时支持多套设备在同一区域使用，从而节省空间成本。HTC Vive 目前仅支持 Windows 系统。

## 第三章 Vaa3D 平台中虚拟现实显示与交互的引入

### 3.1 系统设计

#### 3.1.1 系统设计目标

原始的 Vaa3D 平台中, 虽然已经可以显示并处理 3D 医学图像, 但始终是在 2D 屏幕上操作 3D 的模型/图像, 难以直观地从各个角度观察 3D 图像, 使用鼠标对图像进行平移旋转缩放等操作十分的笨拙, 更难以对图像的某个位置进行精确的定位与进一步的标记操作。

因此我们预想为 Vaa3D 平台引入新型的显示方式, 即将这些 3D 医学图像显示在虚拟现实环境中, 可以更为直观地观察到整个 3D 图像的立体结构, 即便没有实现任何的图像平移旋转操作, 也可以通过在虚拟现实环境中的任意走动与 VR 头盔角度的任意变换, 十分方便的就能观察到 3D 图像各个角度各个维度的图像特征。进一步我们还可以在虚拟现实环境中直接对 3D 图像进行各种直观操作, 例如通过手柄对 3D 图像进行平移旋转缩放等操作, 结合用户走动与 VR 头盔的旋转, 极大地方便了用户对 3D 图像的各个维度与各个角度的观察, 提升了用户的使用体验, 进一步提升了用户的工作效率。

另一方面, 我们预想为 Vaa3D 平台在虚拟现实环境中引入新型的交互方式, 使得之前在 2D 屏幕上通过鼠标键盘能够实现的操作, 例如 追踪单条神经元, 生成一个 marker, 编辑单条 SWC 标注等, 都能够在虚拟现实环境中实现, 甚至某些操作在引入虚拟现实环境中后能得到优化, 进一步提升用户的工作效率。

#### 3.1.2 系统设计架构与细节

本系统设计使用典型的三层架构<sup>[12]</sup>, 即:

- 1) 应用表现层(UI): 主要是指用于与用户进行交互的界面。用于接收用户从 VR

手柄控制器的输入和交互操作。

- 2) 业务逻辑层(BLL):应用表现层和数据访问层之间的桥梁。实现各种功能交互逻辑。功能交互逻辑具体包含：手柄交互处理，数据更新，多种消息响应等等。
- 3) 数据访问层(DAL):与底层数据进行交互。主要实现对数据的增、删、改、查。将底层数据提交给业务层，同时将业务层处理的最新数据保存到底层数据。

本虚拟现实系统的系统架构图如图 3.1 所示。

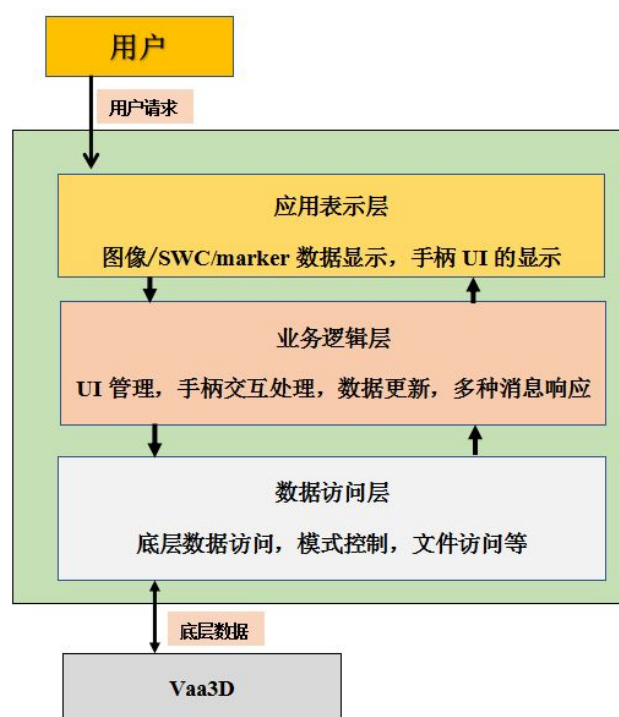


图 3.1 脑图像可视化系统架构图

根据本系统总体功能进行需求分析，我们得到系统各个功能模块设计图，如图 3.2 所示。

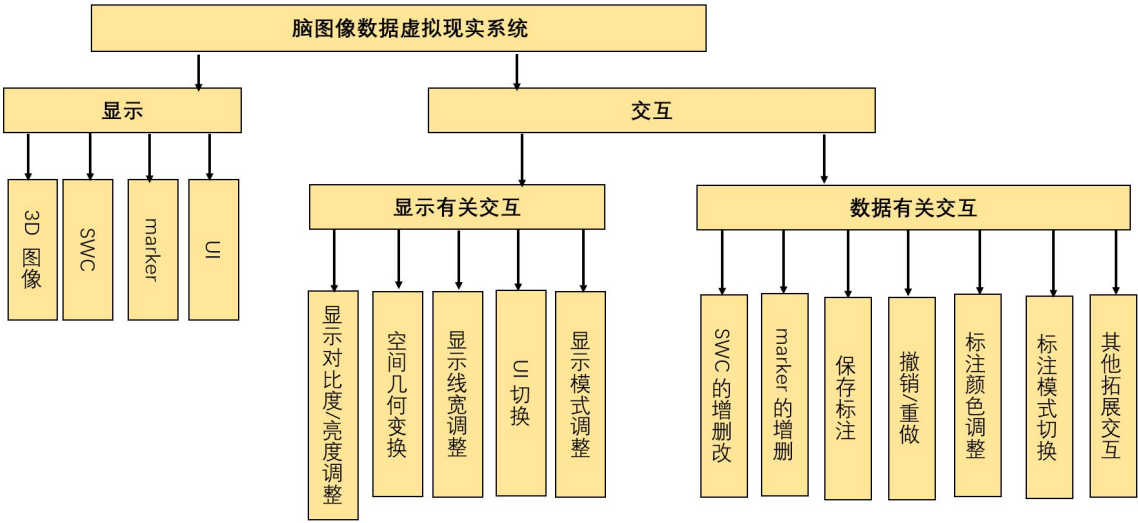


图 3.2 脑图像可视化系统功能模块图

据本系统的功能模块的设计和实际应用，我们重新设计了 HTC Vive 的手柄控制器的 UI。首先我们对手柄控制器的各个按键进行编号排序，方便进行功能说明和用户理解。手柄控制器的编号如图 3.3 所示。

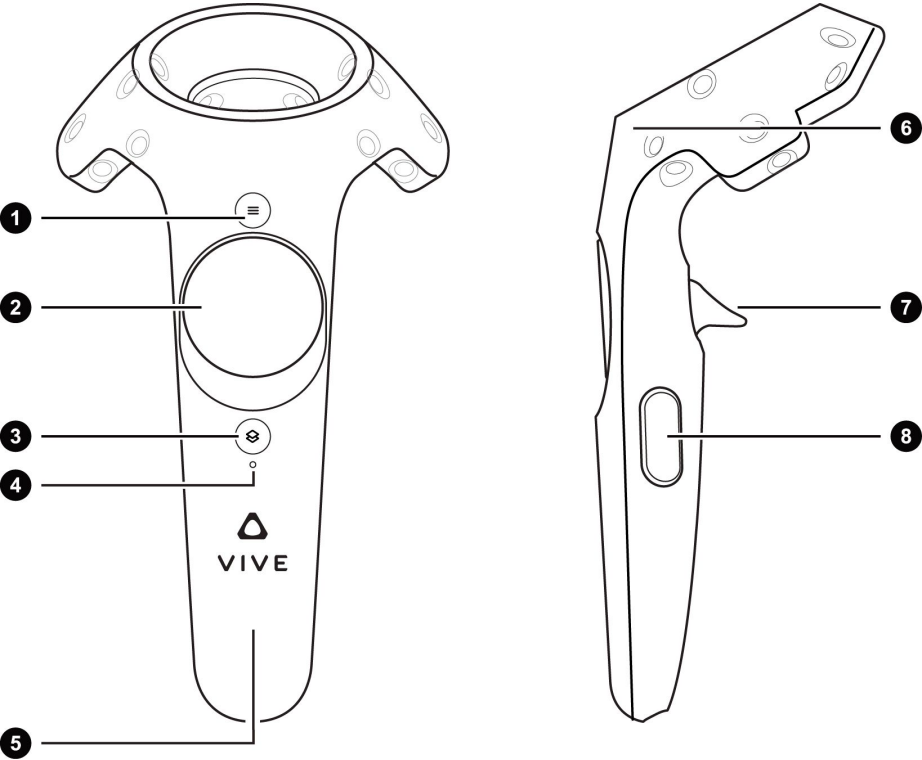


图 3.3 系统自定义手柄控制器按键编号

在正常的虚拟现实环境中，用户都会同时操作两个手柄控制器。HTC Vive

可以同时支持多个(3 个以及更多)手柄控制器,但大多数正常情况下会,本系统中用户只会用到两个手柄控制器,我们将这两个手柄控制器定义为手柄 A 和手柄 B,并且为这两个手柄控制器实现了不同的 UI 和功能。下面介绍这两个手柄控制器的 UI 设计和功能介绍。详细的功能介绍和功能实现将在本章第三节中说明。

#### 手柄控制器 A:

① 保存按钮。用于保存用户在本虚拟现实系统的各种工作,包括神经元重建结果,关键信息标注等等。

② 空间变换平板。用于进行各种平移旋转缩放的空间变换,不同模式下会显示不同的 UI,表示不同的功能,具体包括 图像平移、图像旋转、图像缩放、无变换模式。

③ 系统按钮。HTC Vive 预设按钮,用于用户在虚拟现实环境下随时呼出 VR 系统菜单,可以进行设置和退出等操作。

④ 控制器状态指示灯。HTC Vive 预设功能模块,用于显示当前控制器的状态:正常、未连接、电量不足、充电中等。

⑤ 标注模式显示区域。用于显示当前选中的标注模式,不同模式下会显示不同的 UI,表示不同的功能,具体包括 (i) SWC 标注添加;(ii) SWC 标注删除;(iii) 单个 SWC 标注节点修改;(iv) marker 标注添加;(v) marker 标注删除。

⑥ 标注中心位置显示。此位置在实际应用中会显示一个球体,标明使用次手柄进行标注时的中心位置,即标注 SWC 和标注 marker 都是以此位置为中心点;此外,球体的颜色也标明之后进行标注的颜色。

⑦ 手柄控制器扳机。手柄控制器最重要的交互按键,功能类似于普通 PC 上的鼠标,由按键⑤控制扳机的功能。

⑧ 标注模式切换按钮。用于切换当前的标注模式,每按下一次按钮,即可将当前标注模式切换至下一个模式,对应的,按键⑤的 UI 和按键⑦的功能也会发生变化。

#### 手柄控制器 B:

① 退出按钮。用户按下此按钮即可立即退出虚拟现实环境，即退出本系统。

② 功能确认按钮。用于不同拓展功能下的确认/取消或者增加/减少等。不同模式下会显示不同的 UI，表示不同的功能。

③ 系统按钮。HTC Vive 预设按钮，用于用户在虚拟现实环境下随时呼出 VR 系统菜单，可以进行设置和退出等操作。

④ 控制器状态指示灯。HTC Vive 预设功能模块，用于显示当前控制器的状态：正常、未连接、电量不足、充电中等。

⑤ 拓展功能模式显示区域。用于显示当前选中的拓展功能，不同模式下会显示不同的 UI，表示不同的功能，具体包括（i）对比度/亮度调整；（ii）自动/手动标注模式选择；（iii）Skeleton/Surface 模式选择；（iv）线宽调整；（v）标注颜色调整；（vi）撤销/重做；（vii）自动旋转功能开关；（viii）其他拓展功能。

⑥ 无功能；

⑦ 快捷空间变换交互按钮。持续按下手柄控制器 B 的扳机，即可让整体虚拟空间随着手柄控制器 B 的旋转平移进行相应的旋转平移变换。

⑧ 拓展功能模式切换按钮。用于切换当前的拓展功能模式，每按下一次按钮，即可将当前的拓展模式切换至下一个模式。

### 3.1.3 系统功能总结

根据 2.2 中的系统功能模块图，我们将系统功能分为（i）与显示有关的功能；（ii）与数据有关的功能。表 2.1 中将所有与显示有关的功能列出并简单介绍功能；表 2.2 中将所有与数据有关的功能列出并做了简单的功能介绍。



表 2.1 与显示有关功能简介

功能编号	功能名称	功能简介
1	图像平移	将整体图像（包含所有标注信息）平移至新位置
2	图像旋转	将整体图像（包含所有标注信息）旋转至新角度
3	图像缩放	将整体图像（包含所有标注信息）缩放
4	快捷图像变换	使用手柄 B 扳机快捷地对图像整体进行平移旋转变换
5	对比度调整	调整图像显示的对比度
6	亮度调整	调整图像显示的亮度
7	显示模式切换	切换显示模式 Skeleton/Surface
8	显示线宽调整	调整 Skeleton 模式下曲线的显示宽度

表 2.2 与数据有关功能简介

功能编号	功能名称	功能简介
9	SWC 标注添加	根据手柄轨迹画出连续曲线表示 SWC
10	SWC 标注删除	删除特定 SWC（即某条特定曲线）
11	SWC 标注修改	修改 SWC 中单个节点的位置
12	Marker 标注添加	在手柄 A 中心处生成 marker
13	Marker 标注删除	删除特定 marker
14	标注颜色切换	切换之后生成的标注的颜色
15	标注模式切换	切换当前标注模式自动模式/手动模式
16	保存	保存所有生成的标注（SWC，marker）至本地文件
17	撤销/重做	撤销/重做之前的 SWC 标注操作
18	其他拓展功能	其他的与 Vaa3D 平台模块、插件进行交互的拓展功能

## 3.2 系统实现

### 3.2.1 Vaa3D 中的基本数据结构

#### 1. 3D 神经元图像

Vaa3D 平台支持读取多种格式的图像源文件，并且支持 2D、3D，4D 以及 5D 图像源数据的读取和可视化。在本虚拟现实系统中，所需要显示的大多数为三维图像，常用的三维图像源文件格式包括：Tiff stack (.tif, .tiff); Zeiss LSM

(.lsm);MRC (.mrc); 以及 Vaa3D 平台中输出的图像源文件格式 (.v3draw, .raw, .v3dpbd, .vaa3dpbd)等。以上这些格式的三维图像均能够在本虚拟现实系统中正常高效地显示。

## 2. SWC

神经元形态重建的流程中,以图像为输入,最终得到一个树状结构的数据,这种数据结构一般被存储在 SWC 格式文件(.swc)中。SWC 文件是一种通用的神经元重建结果的存储格式。从数学角度来看,SWC 文件中的数据结构可以描述为节点的集合。此集合中,不允许存在环路,至多存在一条从一个节点到另一个节点的路径(不重复)。SWC 文件中的实际数据结构如表 3.1 所示。

表 3.1 SWC 文件存储的数据结构

n	type	x	y	z	radius	parent
1	1	109.151	183.275	39.745	0.618	-1
2	2	106.652	175.263	41.677	0.618	1
3	3	104.897	171.522	41.317	0.618	2
4	3	103.383	151.460	50.321	0.618	3
5	3	286.439	68.862	77.331	0.618	4
6	2	102.460	153.820	30.095	0.618	2
7	3	114.810	101.314	26.130	0.618	6
8	2	122.050	132.507	101.952	0.618	4

在表 3.1 中,每一行表示 SWC 文件中的一个节点,节点的集合构成了一颗完整的节点树。第一列"n"表示此节点的序号(以 1 为起始点,但不一定表示头节点),SWC 文件中的每个结点的序号都是唯一的不重复的,保证此节点的集合中不会存在环路。第二列"type"表示节点的类型。不同类型的节点拥有不同的意义,例如节点类型为 1(type = 1)表示此节点属于神经元胞体部分,节点类型为 2(type = 2)表示节点属于神经元轴突部分,节点类型为 3(type = 3)表示节点属于神经元树突部分。不同类型的节点在 Vaa3D 平台可视化时以不同颜色来显示

以进行区分。第三至五列"x", "y", "z"表示此节点在空间中的 3D 位置,一般以保留两位小数的浮点数表示。第六列"radius"表示节点的大小,形式上表现为显示整条节点树的 3D Surface 模式时此节点的半径。第七列"parent"表示此节点的父结点的序号(n),每个节点只允许存在一个父结点,但允许存在多个子节点。

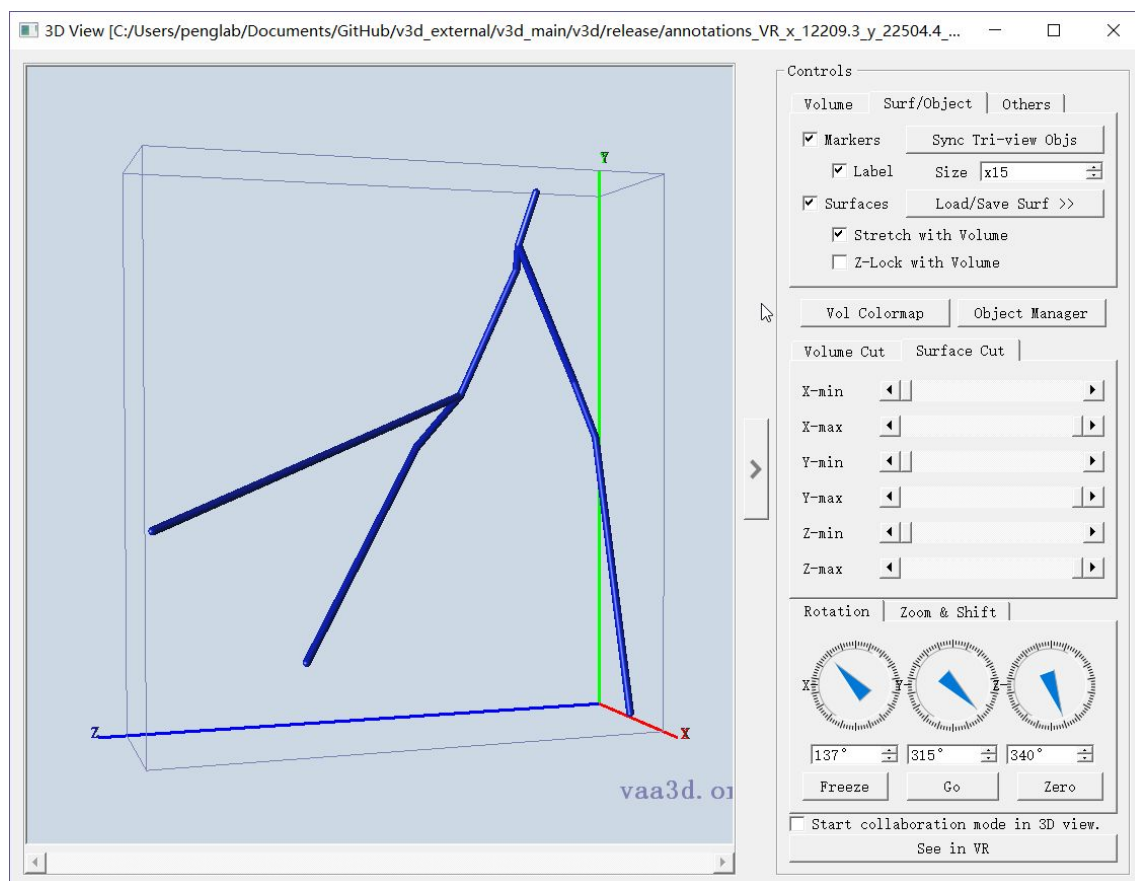


图 3.4 表 3.1 定义的 SWC 在 Vaa3D 中的显示

### 3. Marker

marker 是 Vaa3D 平台一种用于标记的数据结构。此数据结构可以用来标记三维图像中的一些关键信息,比如神经元分叉点,神经元末梢等,也可以用于辅助一些自动示踪(Auto-tracing)算法的结果计算,例如标记自动算法的起始位置、收敛位置以及聚类中心等等。marker 的实际数据结构如表 3.2 所示。

表 3.2 marker 文件存储的数据结构

x	y	z	radius	shape	name	comment	color_r	color_g	color_b
9.61	22.7	22.9	5	1			201	89	127
56.5	54.7	68.3	5	1			255	0	0
314	250	44.5	5	1			0	255	0
179	263	28.4	5	1			0	0	255
104	237	73	5	1			255	255	0
0.05	140	24.7	5	1			255	255	255

在表 3.2 中，第一至三列"x", "y", "z"表示 marker 在空间中的 3D 位置，第四列"radius"表示此 marker 的半径大小，第五列"shape"表示此 marker 的形状，第六列"name"表示此 marker 的名字,第七列"comment"表示针对此 marker 的注释信息，第八至十列"color\_r, color\_g, color\_b"表示此 marker 的颜色信息，即 R、G、B 通道的值，与 SWC 中的 type 属性相同，在某些应用中，不同颜色的 marker 表示不同的意义。

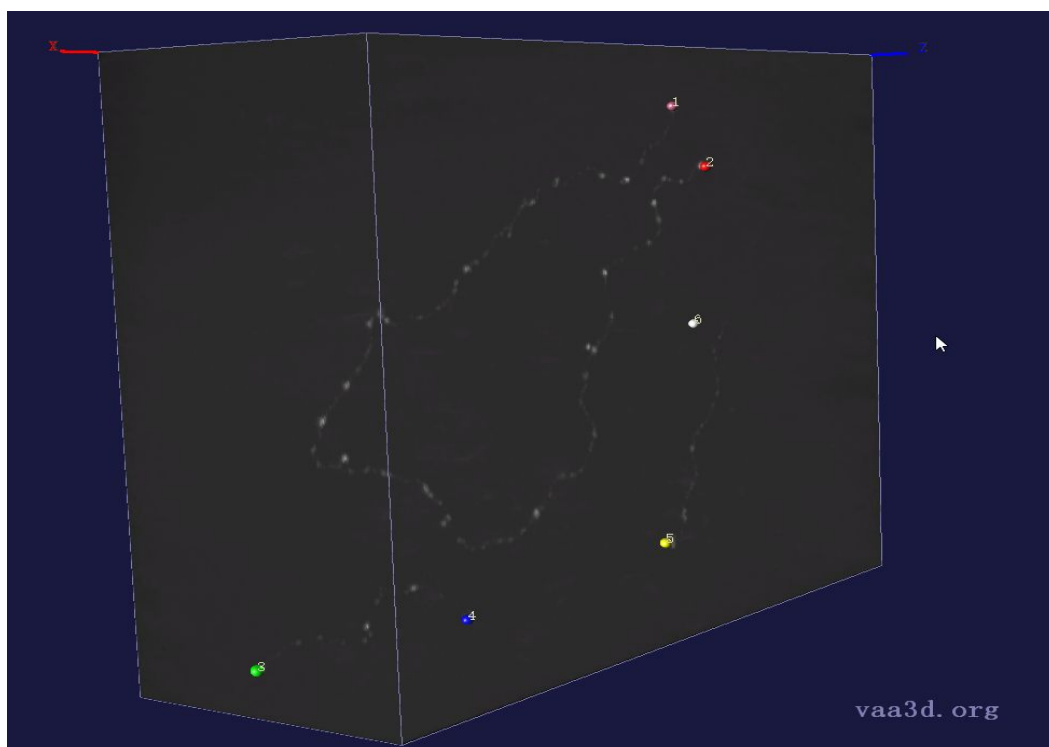


图 3.5 表 3.2 定义的 marker 在 Vaa3D 中的显示

### 3.2.2 3D 图像数据和其他标注的显示实现

#### 1. OpenGL 工具

OpenGL 作为一个十分强大的 3D 图形开发工具，它本身不属于一种编程语言，它是图像硬件的软件接口，介于计算机图形与计算机硬件之间。OpenGL 本身包含七百多个函数<sup>[13]</sup>，本质上属于 3D 空间的图形和模型库。开发人员可以利用 OpenGL 构建 3D 模型和图像，并定义特定的交互操作。OpenGL 具有十足的可移植性，它不依赖于操作系统和硬件环境，并且运行效率高，功能丰富。

OpenGL 的主要功能：模型绘制；基本坐标变换；光照处理；物体着色；纹理映射；动画效果；位图和图像处理；反走样。

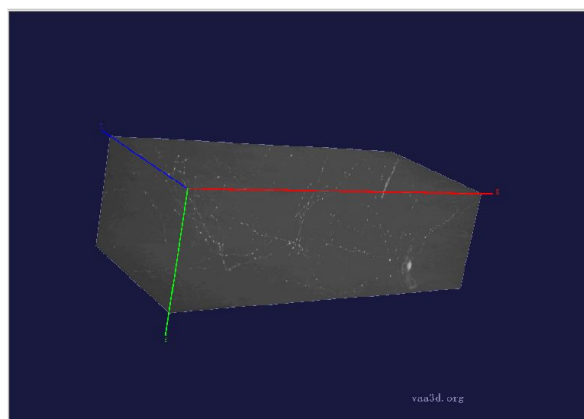
在 OpenGL 中，点是最基本的图形单元，即定点(Vertex)。所有的复杂图形都可以由一组定点的集合描述，顶点间如何连接决定了绘制的图形的类型。一般情况下，OpenGL 中一个定点被画成单个像素(pixel)点。两个顶点的组合构成线段，OpenGL 中的线段与数学定义中的线段不同，是有宽度的。多个线段首尾相连形成的笔和区域就构成了一个多边形，多边形一般由其各个端点来描述。点、线段和多边形构成可 OpenGL 中的基本几何图元<sup>[14]</sup>。

#### 2. 体绘制

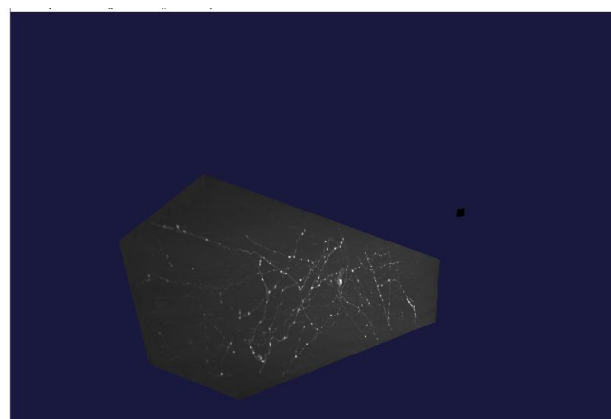
在三维的不规则体绘制 (Volume rendering) 的研究领域，目前有四种常用的可视化算法：光线投射法(Ray Casting)；抛雪球法(Splatting)；错切-变形法 (Shear-warp) 和基于硬件的 3D 纹理映射方法 (Hardware-assisted 3D texture-mapping)<sup>[15]</sup>。本论文使用并改进了其中最为常用的光线投射算法来讲 3D 神经元图像数据在虚拟现实环境中可视化。

光线投射法是一种以图像空间为序的体绘制方法<sup>[16]</sup>。它从图像空间的每一像素出发，按视线方向发射一条射线，这条射线穿过三维数据场，沿着这条射线选择  $K$  个等距的采样点，并由距某一采样点最近的 8 个数据点的颜色值和不透明度值作三次线性插值，求出该采样点的不透明度值和颜色值。再将每条射线上各采样点的颜色值和不透明度值由前向后或由后向前加以合成，即可得到发出该射线的像素点处的颜色值，从而可以在屏幕上得到最终的图像。

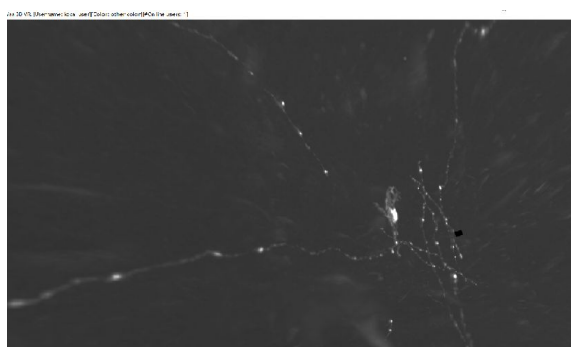
本系统目标是可以实现用户在虚拟现实环境中能够在三维图像数据的内部和外部随意走动的效果，但传统的光线投射法只能应用于一般的 2D 屏幕中的三维对象的渲染，用户的视线和视角始终是在三维对象的外部，因此我们需要对传统的光线投射法进行改进。当用户的眼睛位置，也就是摄像机位置处在 3D 图像数据的内部时，我们依旧是沿视线方向发射出一条射线，但不再选取射入三维数据场的点作为算法计算的起始点，而是直接选取摄像机位置作为算法计算的起始点。简而言之就是，根据摄像机位置发出的射线与图像数据相交的部分，计算出新的立方体，将 3D 图像数据中属于该立方体的区域显示在此立方体中，即可实现人可以走进图像中的效果。具体实现效果如图 3.6 所示。



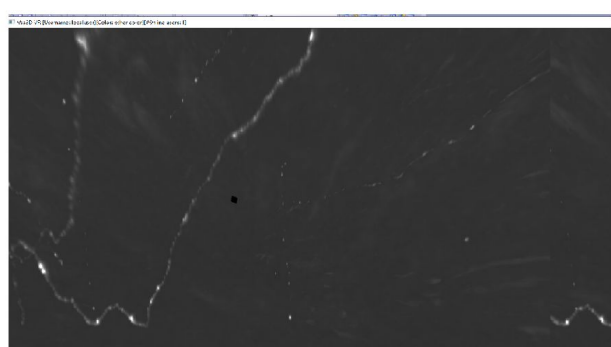
a 图像在 Vaa3D 上的显示效果



b 图像在 VR 中的显示效果 1



c 图像在 VR 中的显示效果 2



d 图像在 VR 中的显示效果 3

图 3.6 体绘制实现效果图示

图 3.6-a 表示 3D 神经元图像在普通 2D 屏幕上的显示效果；图 3.6-b 表示图像显示在虚拟现实环境中时，人眼在图像外部时的整体观察效果；图 3.6-c 表示用户走进图像内部，人眼在图像正中央时朝某一方向的观察效果；图 3.6-d 表示人眼在途 3.6-c 的同样位置朝反方向的观察效果。这几张对比度表明本系统确实可以让用户在虚拟现实环境中走进图像内部进行观察，而且效果良好。

### 3. SWC 显示

在本虚拟现实系统中，SWC 会存在两种不同的显示模式：skeleton 模式和 surface 模式。下面分别介绍这两种显示模式的具体实现：

#### 1) skeleton 模式

使用固定宽度的线段将节点与节点之间连接起来，所有数据中的节点全部连接后最终形成一条连续的曲线，在本系统中的显示效果如图 3.7 所示。此模式下，我们能够更直观地看到整个节点树的拓扑结构，能够更清晰地看到整个神经元结构的分支数和走向，更能轻易分辨出整个节点树中的关键节点。

跟据 SWC 的数据结构，我们用 OpenGL 中的单个顶点表示 SWC 中的每个 node，用顶点的坐标表示 node 在三维空间中的位置，用顶点之间连接而成的线段表示 node 之间的父子关系，用线段的颜色表示 node 的 type。

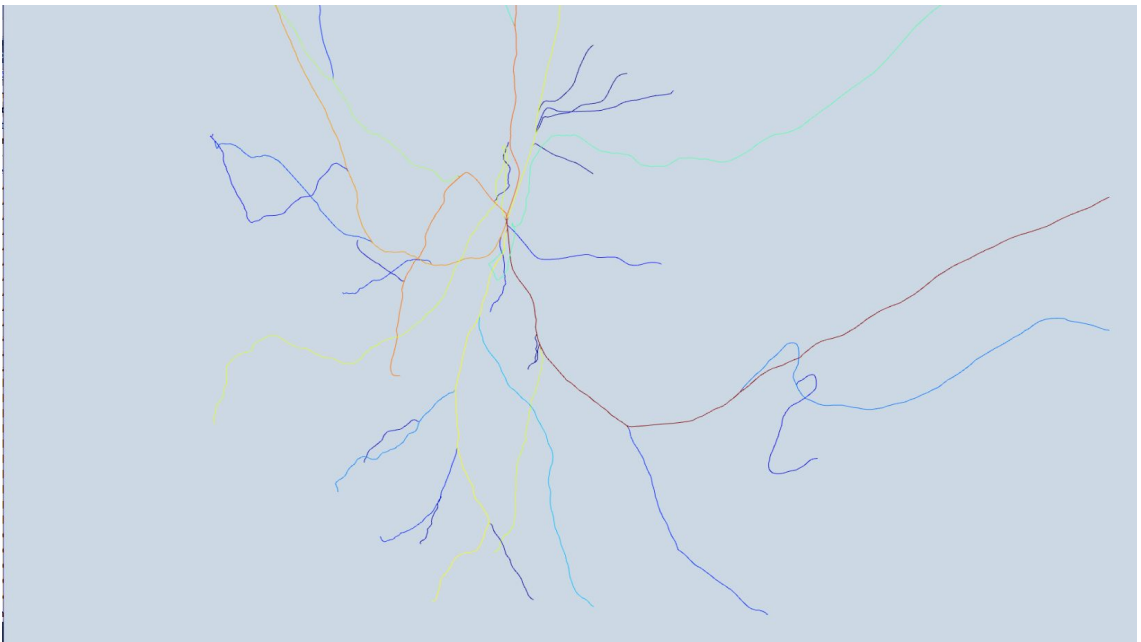


图 3.7 skeleton 模式下 SWC 的显示效果



## 2) surface 模式

此模式着重点在于显示神经元重建结构的 3D 表面信息,能够更加清晰直观地看到 3D 重建结果与实际三维图像数据的拟合程度。将 node 显示成 sphere 将 node 之间的线段显示成 cylinder 针对单个节点,我们会在对应的位置渲染一个颜色半径与该节点的数据结构相同的球体作为该节点的 3D 结构;针对节点之间的连接,我们会根据节点构成的线段的两个端点的半径大小和位置渲染出一个对应半径的圆台来表示。在本系统中此模式的显示效果如图 3.8 所示。

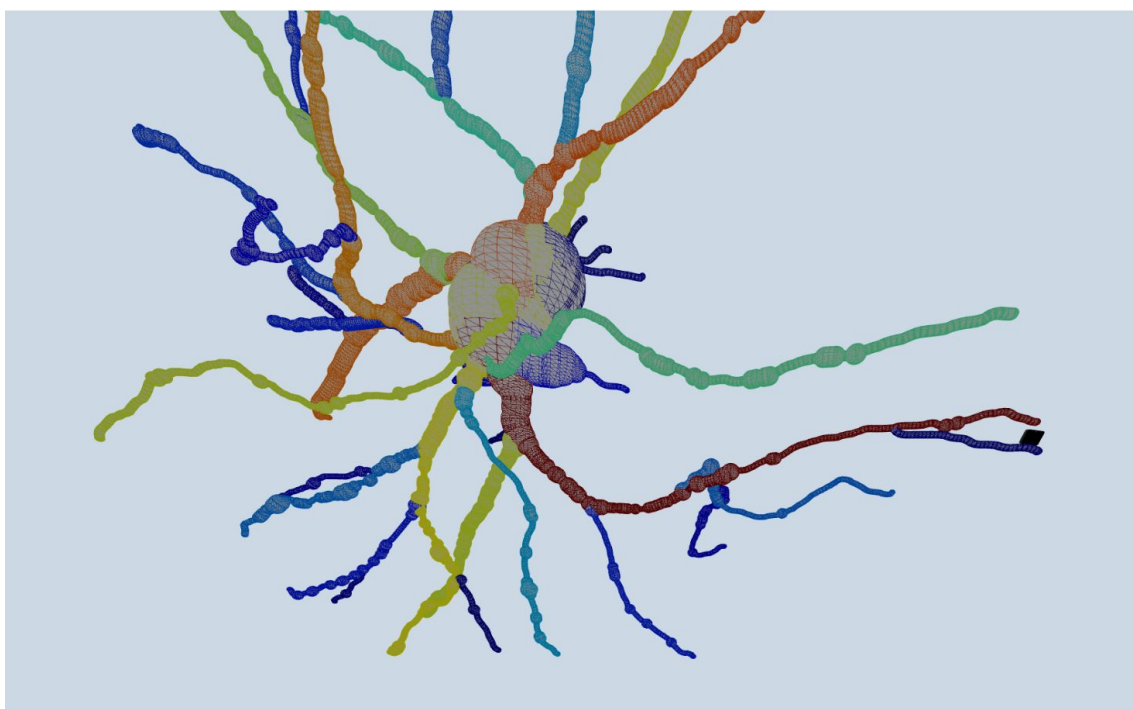


图 3.8 surface 模式下 SWC 的显示效果

#### 4. Marker 显示

在本虚拟现实系统中，marker 的显示方式与 Vaa3D 平台普通的显示方式相似，都是将 marker 作为一个球体显示在对应的空间位置中。marker 的显示效果如图 3.9 所示。



图 3.9 smarker 的显示效果

#### 5. 图像显示优化

根据神经元图像数据的来源和需求不同，神经元的三维图像数据的大小也不相同，较小的可能为  $126*126*126$ ，较大的甚至可能为  $8192*8192*256$  甚至更大。但从用户需求出发，用户始终希望本系统可以以合适的方式显示各种大小的三维图像数据，在每次进入本系统时，都能够以一个合适的角度和大小来观察当前显示的数据。

因此本系统在每次启动时，会根据当前载入的三维图像数据的分辨率( $X*Y*Z$ )选定一个中心锚点(Center position) 和 全局缩放值(Global scale)，再结合用户当前 VR 环境的大小和位置（数据来源于 HTC Vive 的设置参数），通过计算得到一个全局变换矩阵（Global matrix），经过矩阵变换之后将整个 3D Image 图像显示在虚拟现实环境中心的位置，并缩放至一个合适的大小。

### 3.2.3 系统功能的实现

下面根据章节 3.1.3 中的系统功能编号的顺序逐一对这些系统交互功能进行介绍。

#### 1. 图像平移

为用户提供将图像整体平移至另一新的位置的交互操作。

操作具体步骤：

Step1. 按下手柄 A 的按键②，直至切换手柄 A②上方贴图至图 3.10a 所示；

Step2. 触摸手柄 A 的按键②的上下左右部分，即可使图像整体沿对应方向平移。

#### 2. 图像旋转

为用户提供将图像整体旋转至另一角度的交互操作。

操作具体步骤：

Step1. 按下手柄 A 的按键②，直至切换手柄 A②上方贴图至图 3.10b 所示；

**Step2.** 触摸柄 A 的按键②的上下左右部分,即可使图像整体沿对应方向旋转。

### 3. 图像缩放

为用户提供将图像整体按一定大小缩放的交互操作。

操作具体步骤:

**Step1.** 按下手柄 A 的按键②,直至切换手柄 A②上方贴图至图 3.10c 所示;

**Step2.** 触摸柄 A 的按键②的上半部分,即可放大整体图像,反之,触摸下半部分,即可缩小整体图像。

### 4. 快捷图像变换

为用户提供一种更方便的平移旋转操作,用户只需要操作一个按键即可实现图像整体的平移与旋转操作。

操作具体步骤:

**Step1.** 按住手柄 B 的扳机键,即按键⑦,自由旋转拉动手柄 B 即可使整体图像沿着手柄 B 进行平移旋转变换。

### 5. 图像对比度调整

在一般情况下，即不调整任何显示参数的情况下，图像中的某些信号较低的区域在 VR 场景中是不可见或者不太清晰的（同样，在于 Vaa3D 的 2D 屏幕上也是不可见的）。较低信号区域在不同对比度下的 Vaa3D 上的显示对比图如图 3.11 所示。多数情况下，这些亮度/信号强度较为微弱的信号/区域我们可以认定为噪声信号，但同时这些信息在很多时候对于我们来说是有用的，因此我们需要实现在 VR 中也可以清晰看到这些微弱信号的方法。可以通过调整图像显示的对比度和亮度的方法将这些微弱信号清晰地显示出来。

此功能为用户提供了增加/减小显示对比度的交互操作。

操作具体步骤：

**Step1.** 按下手柄 B 的按键⑧，直至切换手柄 B 按键⑤处的贴图至图 3.12a 所示；

**Step2.** 按下柄 B 的按键②的 UI 上的”+/-”部分，即可增加/减小当前图像显示的对比度。

## 6. 亮度调整

此功能为用户提供了增加/减小显示亮度的交互操作。

操作具体步骤：

**Step1.** 按下手柄 B 的按键⑧，直至切换手柄 B 按键⑤处的贴图至图 3.12b 所示；

**Step2.** 按下柄 B 的按键②的 UI 上的”+/-”部分，即可增加/减小当前图像显示的亮度。

## 7. 显示模式切换

此功能为用户提供了切换 SWC 标注的显示模式（Surface-mode or Skeleton-mode）的交互操作。两种显示模式已经在章节 3.2.2 中详细描述。

操作具体步骤：

**Step1.** 按下手柄 B 的按键⑧，直至切换手柄 B 按键⑤处的贴图至图 3.12c 所示；

**Step2.** 当手柄 B 的按键②的 UI 上显示为”ON”，按下按键②即可关闭 Surface mode,也随即打开 Skeleton mode;反之，当手柄 B 的按键②的 UI 显示为”OFF”,按下按键②即可打开 Surface mode,也随即关闭 Skeleton mode。

## 8. 显示线宽调整

此功能为用户提供了增加/减小 SWC 标注的显示线宽的交互操作。

操作具体步骤：

**Step1.** 按下手柄 B 的按键⑧，直至切换手柄 B 按键⑤处的贴图至图 3.12d 所示；

**Step2.** 按下柄 B 的按键②的 UI 上的”+/-”部分，即可增加/减小当前 SWC 标注的显示的线宽。

## 9. SWC 标注添加

手动的对神经元图像信号进行 SWC 标注是生物学家们可能最经常使用的功能，我们需要尽可能的方便/简化用户的操作，因此我们将右手柄的默认模式设置为 SWC 标注模式，只需要按下扳机即可进行操作。当用户在虚拟现实环境下进行神经元重建的工作时，只需要按下扳机并沿着需要标注的位置移动手柄，即可完成对图像信号的描绘。

操作具体步骤：

**Step1.** 按下手柄 A 的按键⑧，直至切换手柄 A⑤上方贴图至图 3.13a 所示；

**Step2.** 持续按住手柄 A 的扳机键，即按键⑦，即可沿着手柄 A 的移动轨迹生成连续曲线。

## 10. SWC 标注删除

在实际的神经元重建工作中，必定会产生各种各样的误差操作或者错误操作，因此我们需要实现纠错的功能。我们为此系统设计了两种不同的纠错功能：

（i）删除模式，此模式适用于单条 SWC 标注误差过大或者位置错误需要整条删除的情况；（ii）移动模式，此模式适用于单条 SWC 标注中只有单个或者少量节点误差过大的情况，仅修改单个节点位置即可。

此功能为用户提供了删除单条 SWC 标注的交互操作。

操作具体步骤：

**Step1.** 按下手柄 A 的按键⑧，直至切换手柄 A⑤上方贴图至图 3.13b 所示；

**Step2.** 按下手柄 A 的扳机键，即按键⑦，即可删除手柄一定范围内距离最近的一条 SWC 标注。

## 11. SWC 标注修改

此功能为用户提供了修改 SWC 标注中某个节点位置的交互操作。

**Step1.** 按下手柄 A 的按键⑧,直至切换手柄 A⑤上方贴图至图 3.13c 所示;

**Step2.** 持续按住手柄 A 的扳机键,即按键⑦,即可选取手柄一定范围内距离最近的一个 SWC 节点;

**Step3.** 选取状态下移动手柄 A 即可将该节点位置动态修改为手柄 A 当前中心位置。

## 12. Marker 标注添加

实际的重建工作中,专家们经常会需要用到 marker 来对关键节点或者图像信号中关键位置进行标记,而且有些自动/半自动重建算法也依赖于专家们标记的 marker 来产生更为精确的重建结果。因此,我们需要在虚拟现实环境下同样实现 marker 的增删操作。在新增 marker 模式下,按下右手柄扳机即可在当前手柄位置生成一个球作为 marker 显示在对应位置;在删除 marker 模式下,在想要删除的 marker 的附近按下右手柄扳机即可删除对应 marker。

此功能为用户提供了新增 marker 标注的交互操作。

操作具体步骤:

**Step1.** 按下手柄 A 的按键⑧,直至切换手柄 A⑤上方贴图至图 3.13d 所示;

**Step2.** 按下手柄 A 的扳机键,即按键⑦,即可在当前手柄 A 中心位置处生成并显示一个球(marker)。



### 13. Marker 标注删除

此功能为用户提供了删除 marker 标注的交互操作。

操作具体步骤：

**Step1.** 按下手柄 A 的按键⑧，直至切换手柄 A⑤上方贴图至图 3.13e 所示；

**Step2.** 按下手柄 A 的扳机键，即按键⑦，即可删除手柄一定范围内距离最近的一个 marker。

### 14. 标注颜色切换

此功能为用户提供了改变标注颜色（包括 SWC 和 marker）的交互操作。当前系统设定为颜色在白-黑-红-蓝-紫之中循环。

操作具体步骤：

**Step1.** 按下手柄 B 的按键⑧，直至切换手柄 B 按键⑤处的贴图至图 3.14a 所示；

**Step2.** 按下按键②即可切换标注颜色至下一颜色。

### 15. 标注模式切换

此功能为用户提供了切换当前标注模式（手动模式/半自动模式）的交互操作。手动模式即为完全按照用户的手柄的移动轨迹进行 SWC 标注的模式，半自动模式的概念与原理将在第四章中进行详细描述。

操作具体步骤:

**Step1.** 按下手柄 B 的按键⑧, 直至切换手柄 B 按键⑤处的贴图至图 3.14b 所示;

**Step2.** 当手柄 B 的按键②的 UI 上显示为"ON", 按下按键②即可关闭半自动模式, 切换为手动模式;反之, 当手柄 B 的按键②的 UI 显示为"OFF", 按下按键②即可打开半自动模式。

## 16. 保存

此功能为用户提供了保存所有标注至本地文件的交互操作。所有的 SWC 标注都会被合并之后保存在当前系统工作目录下, 命名格式为"annotations\_VR\_"+"图像源数据名称"+"当前时间"+" .swc", 所有的 marker 都会被保存在当前系统工作目录下, 命名格式为" annotations\_VR\_"+"图像源数据名称"+"当前时间"+" .marker"。当前工作进行到一定阶段, 可以随时保存, 下一阶段工作开始后就可以重新载入上次工作保存下来的文件, 继续展开后续的工作。

具体操作步骤:

**Step1.** 随时按下手柄 A 的按键①即可保存当前所有标注。按键①的 UI 如图 3.14c 所示。

## 17. 撤销/重做

撤消重做是一个文档编辑或者其他编辑时常用的基本功能, 我们在本系统中, 也实现了 SWC 标注操作的撤消重做功能。撤消重做功能一般有两种方案: 恢复数据和恢复操作。

本系统中选择的是恢复数据的方案。具体方案是当用户每次进行 SWC 标记操作之后，将旧数据保存至一个单独的固定大小的栈(undo\_stack)中，系统限制最大的撤销次数为 5 次，达到最大撤销次数之后，每次 SWC 标注操作产生后，将栈中最旧的数据删除，再将此次数据保存至栈中；当用户每次进行撤销操作时，将旧数据保存至一个单独的固定大小的栈（定义为 redo\_stack）中，从撤销的栈(undo\_stack)中取出最新的一份数据当作当前整个工作环境的数据，redo\_stack 的大小限制也是 5；当用户每次进行重做操作之后，从重做的栈(redo\_stack)中取出一份最新的数据当作整个工作环境的数据，然后清空整个撤销的栈(undo\_stack)。

#### **操作具体步骤：**

**Step1.** 按下手柄 B 的按键⑧，直至切换手柄 B 按键⑤处的贴图至图 3.14d 所示；

**Step2.** 按下手柄 B 的按键②的”UNDO”部分，即可撤销上一次 SWC 标注操作(如果 undo\_stack 存在并可用)；按下手柄 B 的按键②的”REDO”部分，即可恢复上一次的撤销操作(如果 redo\_stack 存在并可用)。

#### **18. 其他拓展功能**

本系统中还定义了多种与 Vaa3D 平台其他模块或者插件整合的系统功能，例如：为了方便用户在 Vaa3D 平台的 Terafly 模块中使用本系统，定义了在本系统中直接调用 Terafly 模块的切换区域与缩放的功能。这些功能非系统必要功能，因此本论文中不做详细介绍。

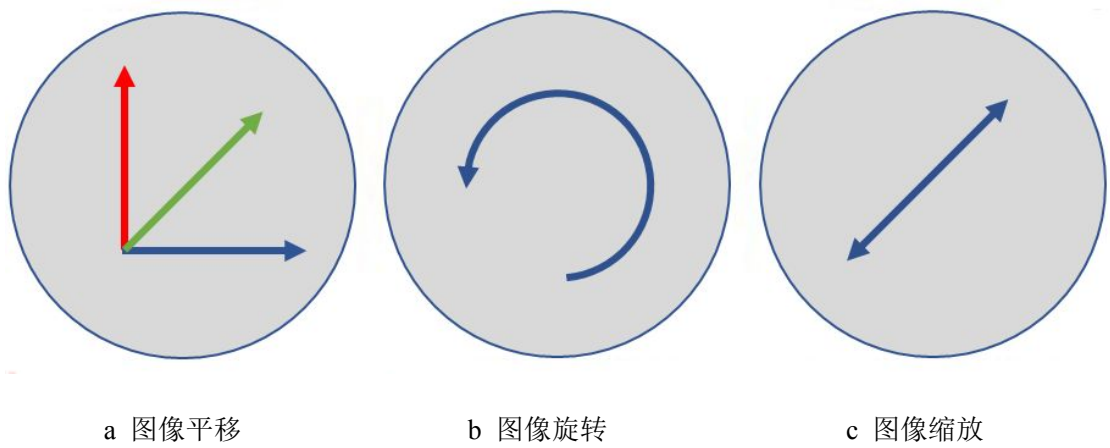


图 3.10 系统功能编号 1-3 对应 UI

图 3.11 信号较低区域不同对比度下的显示对比图

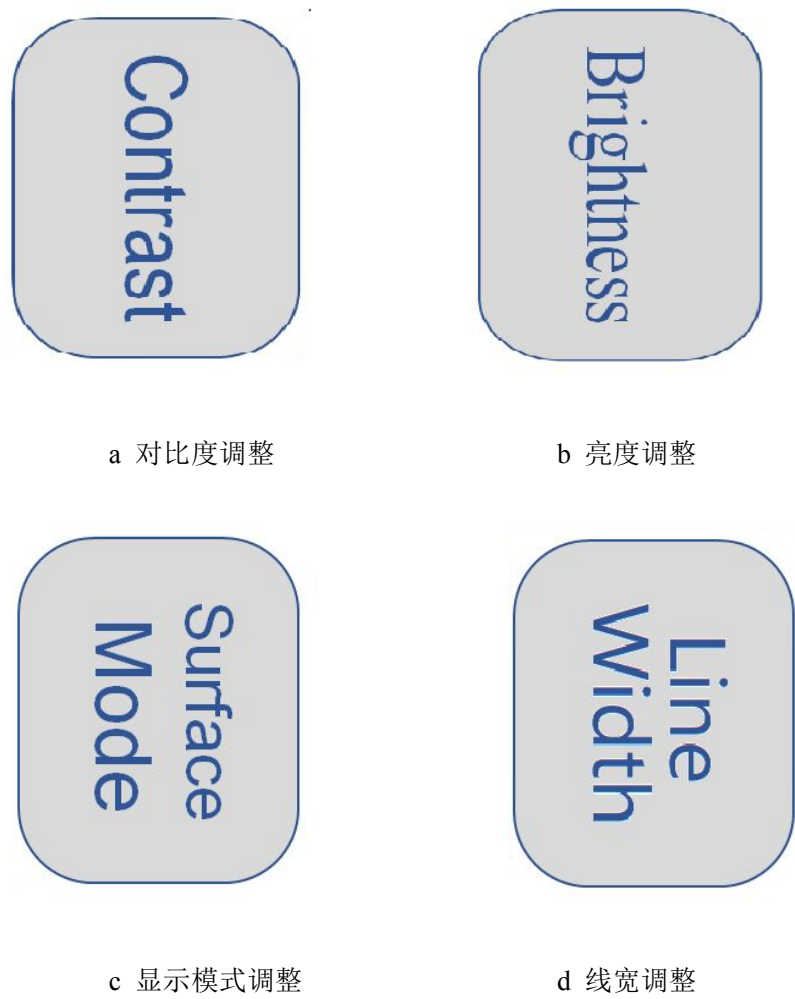


图 3.12 系统功能编号 5-8 对应 UI

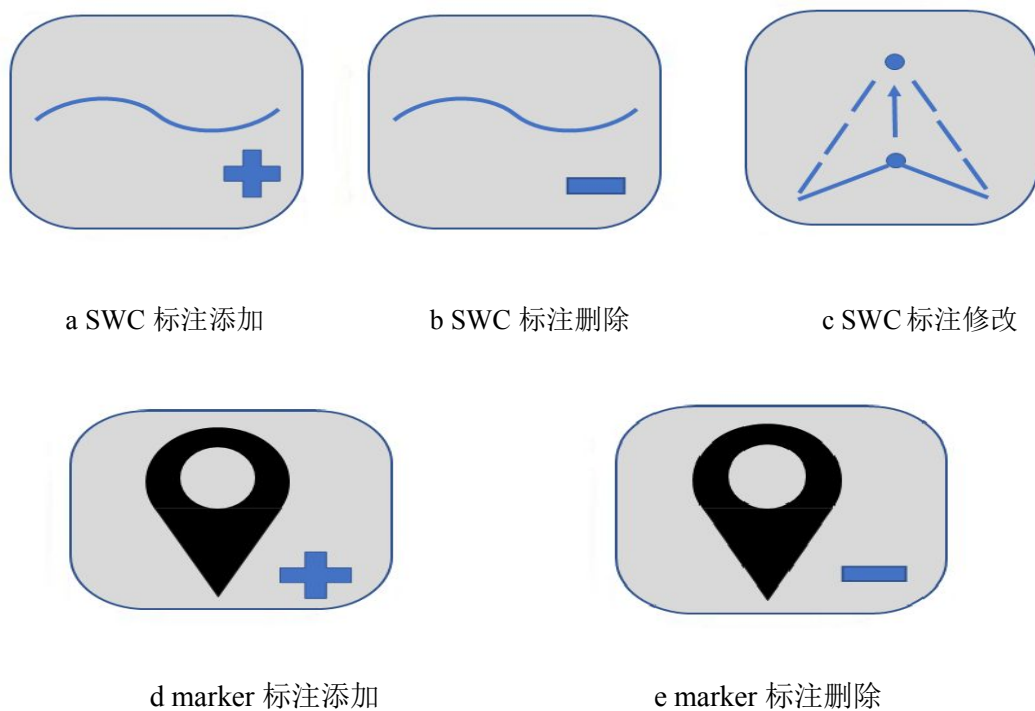


图 3.13 系统功能编号 9-13 对应 UI



图 3.14 系统功能编号 14-17 对应 UI

## 第四章 虚拟现实环境中的智能交互方式

### 4.1 智能交互方式的定义

#### 4.1.1 神经元形态重建的三种交互方式

传统的基于 Vaa3D 平台的进行神经元形态重建的工作流程中，一般存在三种工作方式，分别是 手动重建方法(Manual tracing)、自动重建方法(Auto tracing)和半自动重建方法(Semi-auto tracing)<sup>[17]</sup>。手动重建方法指的是完全依靠生物专家或研究者在屏幕上根据所见到的图像信号进行追踪与标注的方法。

手动的方法完全依赖人工操作，虽然精确性得到保障，但会耗费用户大量的精力与时间。一个小鼠的全脑的所有神经元的标注可能需要多个生物专家协同工作数周甚至数月时间才能完成<sup>[17]</sup>。

与之相反，自动重建方法指的是完全不依赖任何人工干预和操作，基于计算机的自动重建算法，自动地根据图像信号来进行神经元形态重建的方法。自动的重建方法省去了大量的用户的精力和时间，利用计算机自动地生成神经元的重建结果的速度远远快于人工的重建的速度，但由于当前自动重建算法的研究的限制，很难出现一种自动重建算法能够兼顾重建速度和重建结果的正确性，也缺少能够良好应用于多种复杂图像的自动重建算法。

半自动重建方法指的是在计算机自动计算重建结果的输入，输出以及计算过程中施以一些人为的干预的重建方法。这些人为的干预可能包括：(i)在图像数据中人为地标注出重建算法的起始点或者收敛点；(ii)对重建算法的重建结果进行人为的筛选或者修剪；(iii)对重建算法的中间结果进行人为的筛选等等。

半自动的重建方法相比完全自动的重建方法来说，重建结果的准确性更好，更符合生物学专家对于重建结果的期望，而且相比纯手动的重建方法，速度和效率都有很大程度的提升。但仍然需要一定的人工干预的特点决定了半自动的重建方法无法适用于大批量神经元数据的重建工作，需要针对每个单独的神经元数据进行一定的人为干预，而不是像自动重建方法可以自动遍历所有神经元

数据从而得到重建结果。

### 4.1.2 虚拟现实环境下的智能交互方式

本系统作为一个脑图像数据的可视化和交互系统，在第三章中已经描述了使用手柄控制器进行手动重建的工作流程。与此同时，我们也设计为用户提供一种虚拟现实环境中的智能交互方式，可以为用户实现一种更为方便高效的半自动重建方法。

该智能交互方式的设计方案为：用户仍然使用手柄控制器来进行 SWC 标注操作，但在智能交互模式下，即使用户使用手柄控制器随意操作，比如沿着图像信号速度很快的描过去，系统会根据这条比较随意的轨迹与图像信号计算出一条更加贴合图像信号的曲线。在智能交互模式下，用户的输入是一条粗略的节点数较少的折线，输出则是一条自动贴合图像信号且重新采样的 SWC。

Virtual Finger 算法是 Vaa3D 平台中自带的一种用于生成三维空间中的点，曲线和感兴趣区域（ROI，Region of Interest）的算法<sup>[18]</sup>。生物学家在 Vaa3D 平台上进行的标注操作都是基于 Virtual Finger 算法实现的。本论文中，我们设计并实现了一种改进的适用于虚拟现实环境的 Virtual Finger 算法，来实现我的的智能交互方式。

下面首先介绍 Virtual Finger 算法。

## 4.2 Virtual Finger 算法

Virtual Finger 算法以鲁棒有效的方式生成三维空间中的点，曲线和感兴趣区域（ROI，Region of Interest）。只要这些对象在 2D 的显示屏幕中是可见的，只需单击鼠标或者移动鼠标（或者其他类似的输入设备（如数字化笔或触摸屏）的等效操作）即可使 Virtual Finger 算法在图像数据中计算出并显示其在三维空间中的位置。VF 技术允许对复杂 3D 图像内容进行即时和随机顺序的探索，就像我们的真实手指使用单击或笔画探索真实的 3D 世界来定位 3D 对象。

Virtual Finger 算法是一系列满足 3D-WYSIWYG（"What you see in 2D is

what you get in 3D") 基本特征的算法的集合, 将用户在计算机屏幕的 2D 平面中的输入映射到生物实体(例如, 细胞, 神经元等)的 3D 图像数据在三维空间中的位置<sup>[18]</sup>。Virtual Finger 算法得到的三种对象对应于典型光学显微镜图像中的重要结构: (i) 3D 点可以标记标记细胞或蛋白质的位置; (ii) 3D 曲线可对应于一般血管类结构, 如微管, 血管, 支气管树 或神经突触等; (iii) 3D ROI 可以突出特定细胞群或脑室。

Virtual Finger 算法进行神经元重建的具体步骤<sup>[18]</sup>:

- 1) 获取用户在 2D 屏幕上的输入, 即一条由鼠标移动轨迹定义的曲线;
- 2) 将用户输入的曲线重采样得到  $N$  条从屏幕射向 3D 图像数据的射线;
- 3) 使用 PPA 算法<sup>[18]</sup>计算得到 3D 图像数据中的  $N*m$  个点( $m$  表示每条射线在 3D 图像数据中可能命中的 3D 点的个数);
- 4) 根据步骤 2 中的射线的角度对 3D 图像数据进行 2D 平面投影(一般以 XY/XZ/YZ 为投影平面)
- 5) 以这  $N*m$  个点作为输入, 根据步骤 4) 得到的投影数据, 使用 fast marching 算法<sup>[19]</sup>计算每两条射线之间的最短测地路径;
- 6) 最终得到多条从起始射线到达最终射线的局部最短路径, 排序后得到一条最短的全局最短路径。
- 7) 根据全局最短路径生成对应的 3D 曲线, 也就是 SWC 曲线, 平滑后显示在 2D 屏幕中。

### 4.3 在 VR 中的改进的 Virtual Finger 算法

传统的 2D 屏幕获得的用户输入只能是一系列二维的坐标( $x,y$ ), 需要通过算法计算得到/估计用户输入的二维坐标在三维图像数据中的对应的三维位置( $X,Y,Z$ ), 但在虚拟现实环境下, 得到的用户输入始终是一个三维的坐标, 而且能够与三维图像数据中的位置对应起来。因此, 我们可以省去 Virtual Finger 算法流程中计算 3D 点的过程(步骤 1~3)。也就是说我们的首次输入从屏幕的一系列 2D 点( $X, Y$ ), 变成了 VR 空间中的一系列 3D 点( $X, Y, Z$ )。



在 Virtual Finger 算法对 3D 图像数据做 2D 平面投影的步骤中,由于只能确定两个维度的坐标,另一个维度的坐标无法确定,只能做一个方向的投影,例如 X, Y 坐标确定, Z 坐标不确定,所以只能做 XY 平面的投影。但在虚拟现实环境下,三个维度的坐标都可以确定,因此我们可以做任意方向的投影,包括 XY 平面, XZ 平面和 YZ 平面。

考虑 VR 空间中可能会额外存在界外的情况,我们需要将不属于三维图像数据包围盒内的点从输入的曲线中剔除掉,以防止算法出现崩溃或无法收敛的问题。

虚拟现实环境中使用智能交互进行神经元重建的具体步骤:

- 1) 获取用户在虚拟现实环境下的输入,即用户交互操作发生时的手柄控制器的中心 3D 位置( $X_c, Y_c, Z_c$ )最终形成的连续曲线;
- 2) 对这条曲线进行重采样得到 N 个 3D 点;
- 3) 对 N 个 3D 点进行筛选,剔除不属于三维图像数据包围盒内的点,最终得到 N'的 3D 点;
- 4) 对 3D 图像数据进行 2D 平面投影(默认初始 XY 为投影平面)
- 5) 以这 N'个 3D 点作为输入,根据图像数据,使用 Virtual Finger 算法的步骤 5&6 得到一条连续曲线;
- 6) 以上步骤中得到的连续曲线为输入,将投影平面依次改为 XZ 平面、YZ 平面,重复步骤 2~5,最终得到一条更加贴合图像数据精确度更高的连续曲线作为重建结果。

算法流程图如图 4.1 所示。

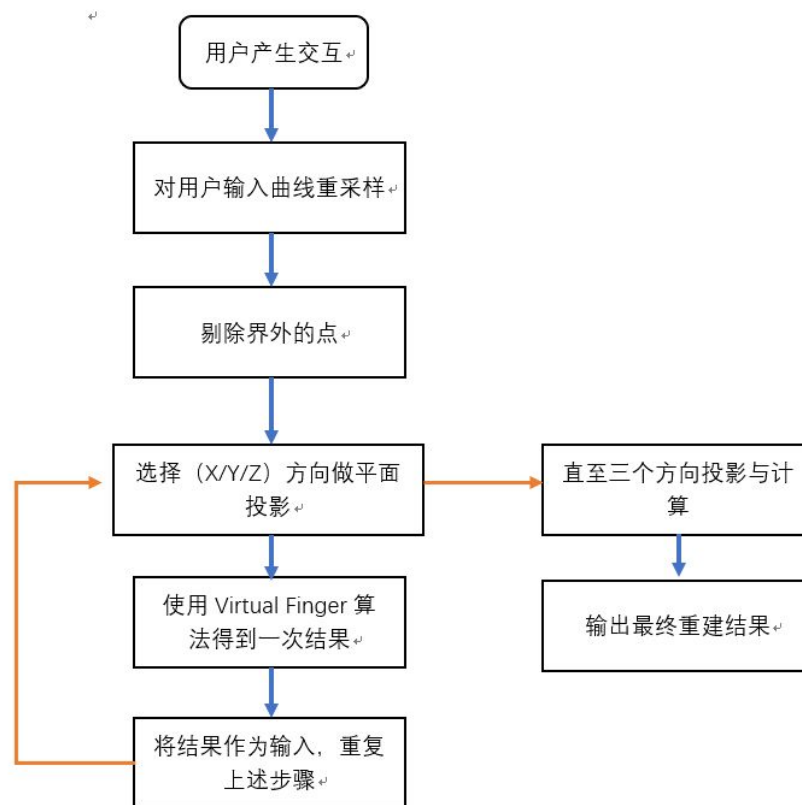


图 4.1 改进的 Virtual Finger 算法流程图

## 第五章 可视化系统应用实例与分析

### 5.1 手动与半自动重建方法的比较与分析

本章节使用同一个 3D 神经元图像数据文件，在 2D 屏幕上的显示效果如图 5.4 所示，将该文件载入可视化系统显示，分别使用手动标注模式和半自动标注模式对同一数据的同一结构进行标注。我们总共进行三次实验：第一次实验使用手动标注模式快速（5 秒内）描绘图像信号，观察所得到的重建结果，如图 5.5 所示；第二次实验使用半自动标注模式快速（5 秒内）描绘图像信号，观察到的重建结果如图 5.6 所示；第三次实验使用手动标注模式仔细缓慢（45 秒左右）地描绘图像信号，观察到的重建结果如图 5.7 所示。得到的三种重建结果放置在同一位置上进行比较的结果如图 5.8 所示，其中黄色表示第一次实验结果，红色表示第二次实验结果，白色表示第三次实验结果。

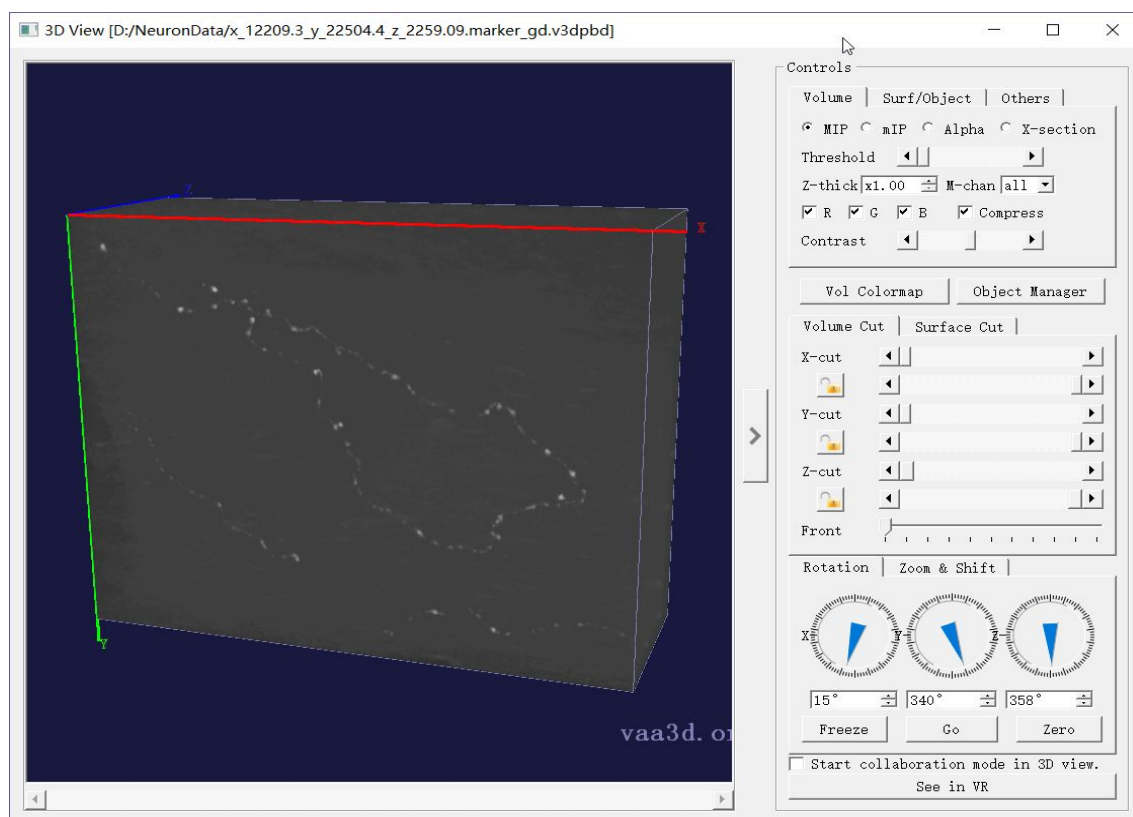


图 5.4 测试数据 1 在 2D 屏幕显示效果

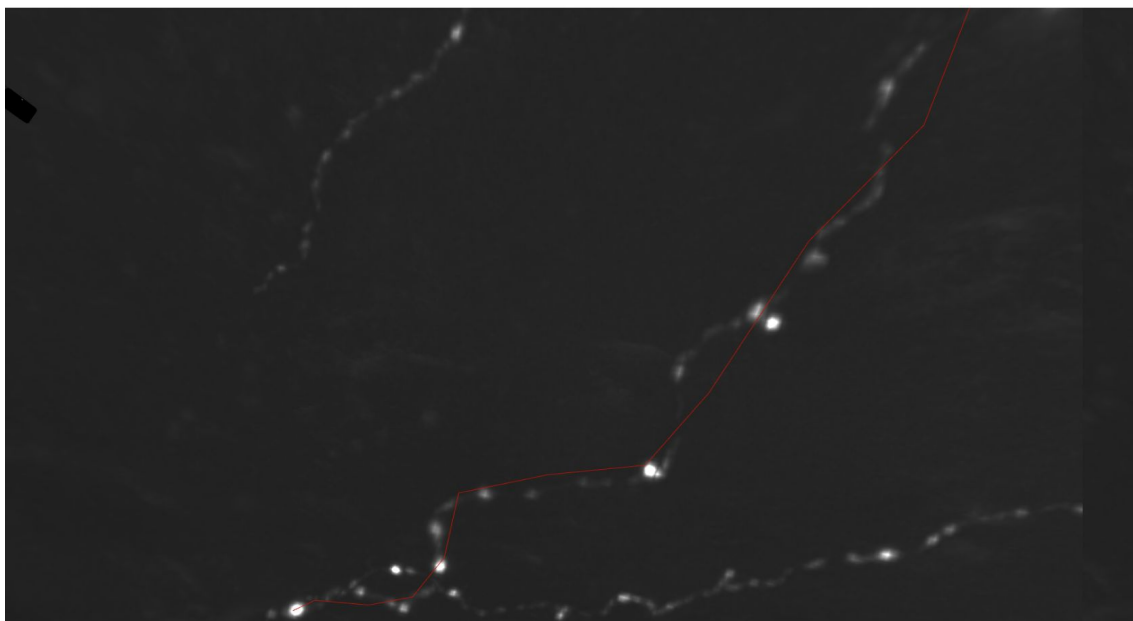


图 5.5 手动标注模式快速标注结果

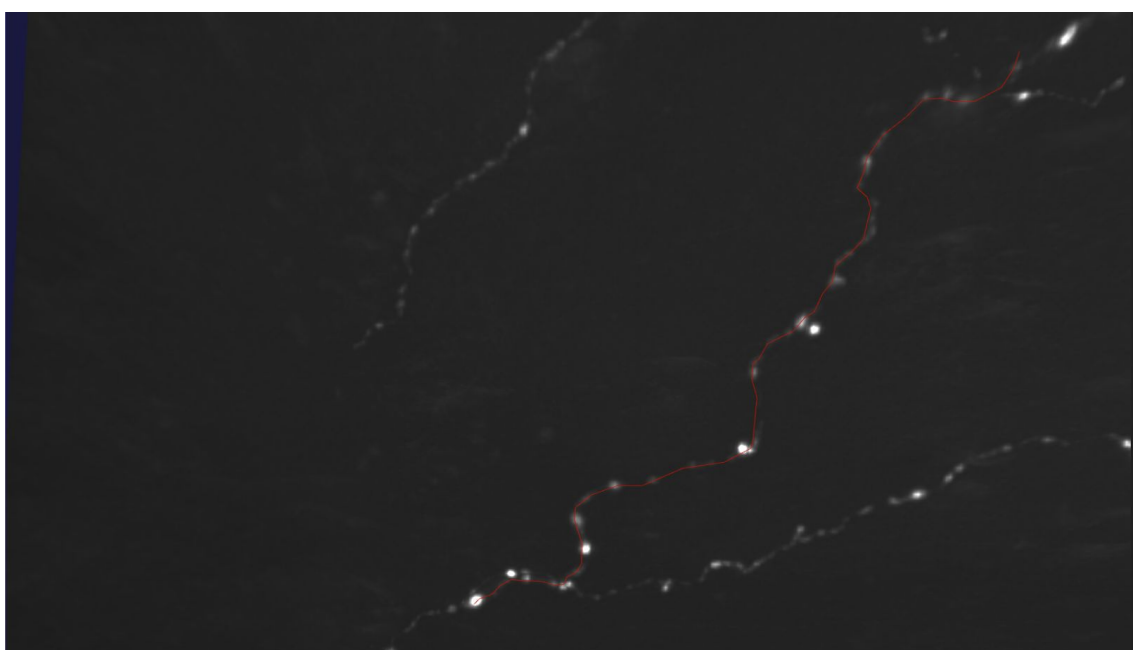


图 5.6 半自动标注模式快速标注结果

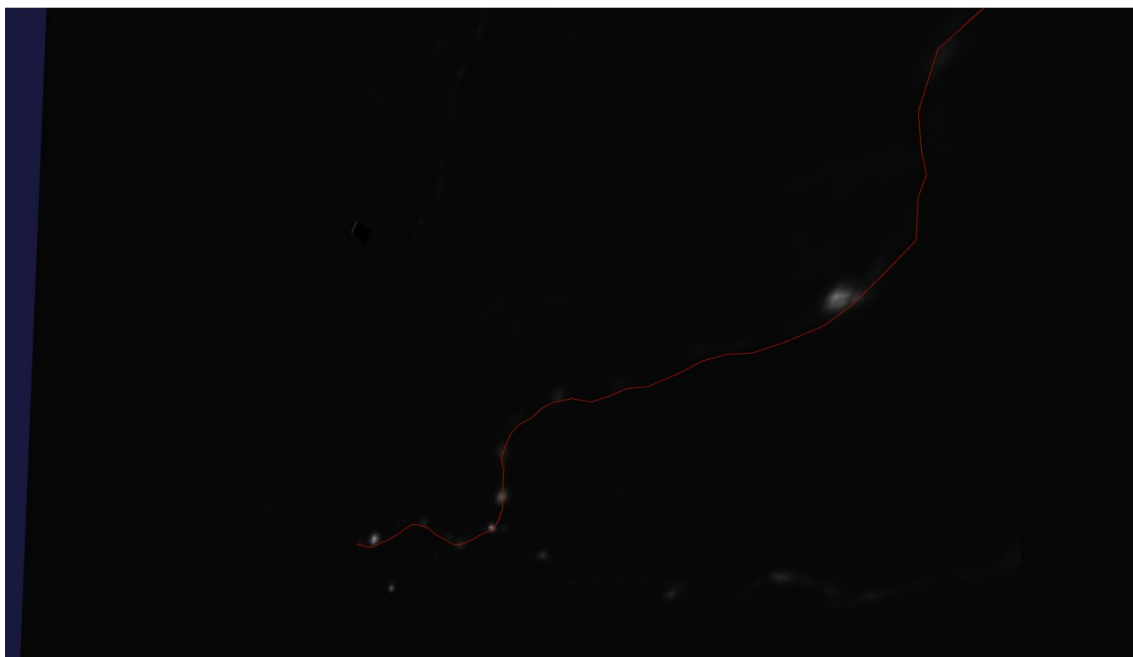


图 5.7 手动标注模式缓慢标注结果



图 5.8 三种实验结果对比

从对比图 图 5.8 中可以看出，同样是快速描绘得到的结果，半自动标注模

式下得到的重建结果（红色）明显更贴合实际图像信号，手动标注模式下的结果（黄色）误差较大且曲线不够平滑；而是用手动标注模式缓慢认真描绘得到的结果（白色）也十分贴合实际图像信号，误差很小，与红色结果误差也很小，但耗费时间是半自动标注模式的 9 倍。以上足以说明，半自动标注模式相比手动标注模式更加节省时间，更加快捷，而且具有足够高的精确度。

## 5.2 系统中的重建与普通重建方法的比较与分析

本章节使用同一个 3D 神经元图像数据文件，在 2D 屏幕上的显示效果如图 5.9 所示，分别使用普通的 2D 屏幕上的手动重建方法和本系统中的手动重建方法对同一数据进行标注。第一次实验使用 Vaa3D 平台中的手动重建方法，得到的重建结果如图 5.10 所示；第二次实验使用本系统中的手动重建方法进行标注，得到的重建结果如图 5.11 所示。两次试验得到的重建结果叠加在一起的对比图如图 5.12 所示，其中黄色代表第一次实验结果，红色代表第二次试验结果。

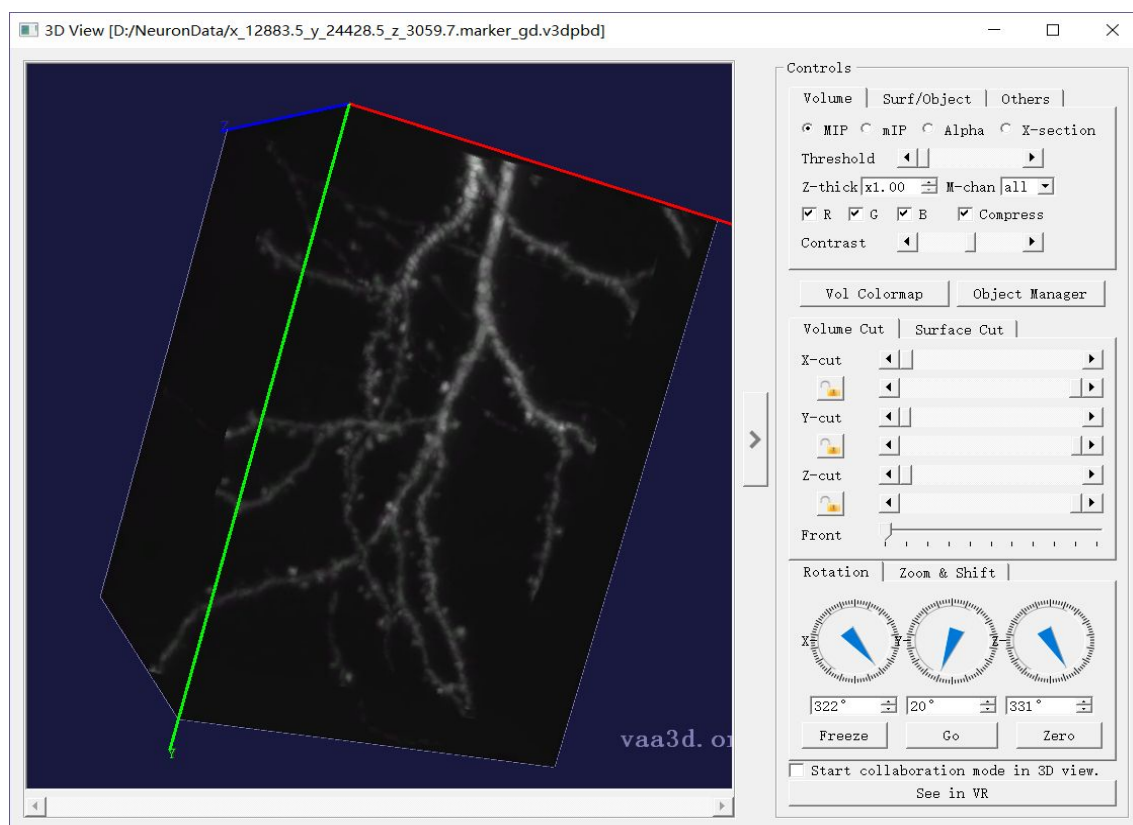


图 5.9 测试数据 2 在 2D 屏幕显示效果

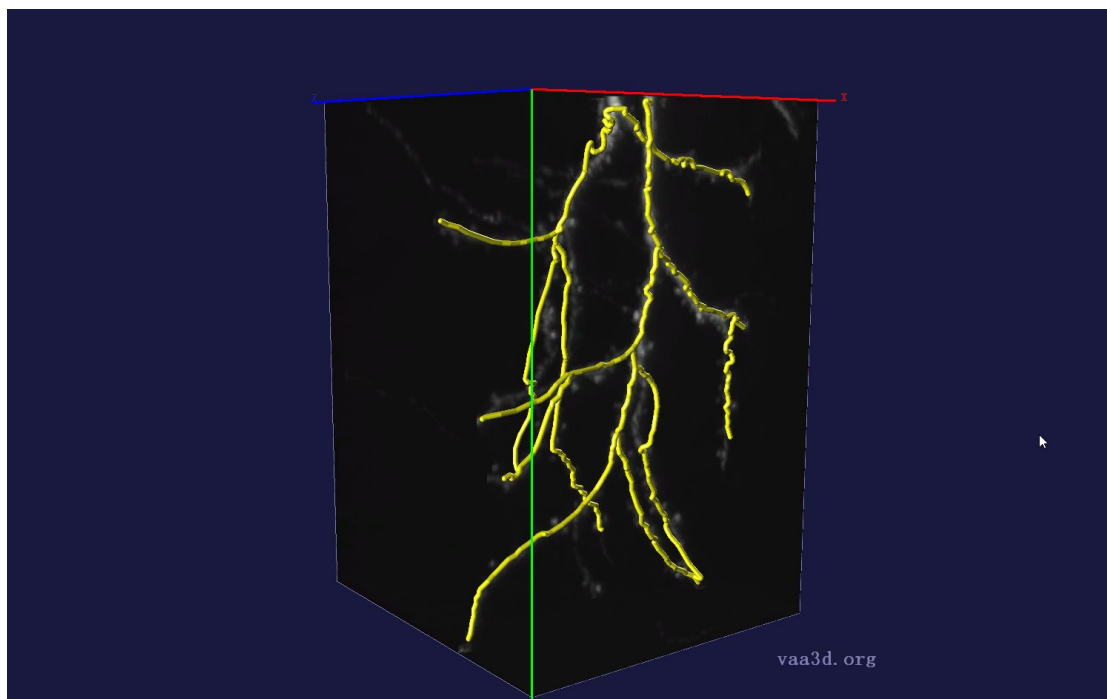


图 5.10 2D 屏幕手动标注结果

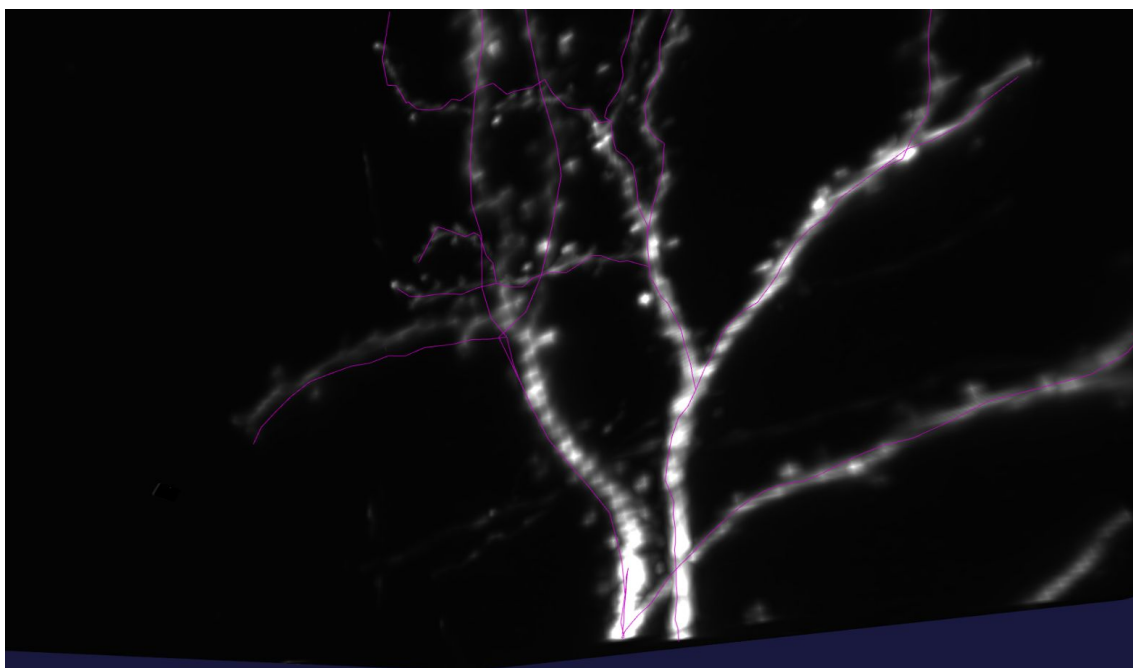


图 5.11 可视化系统中手动标注结果

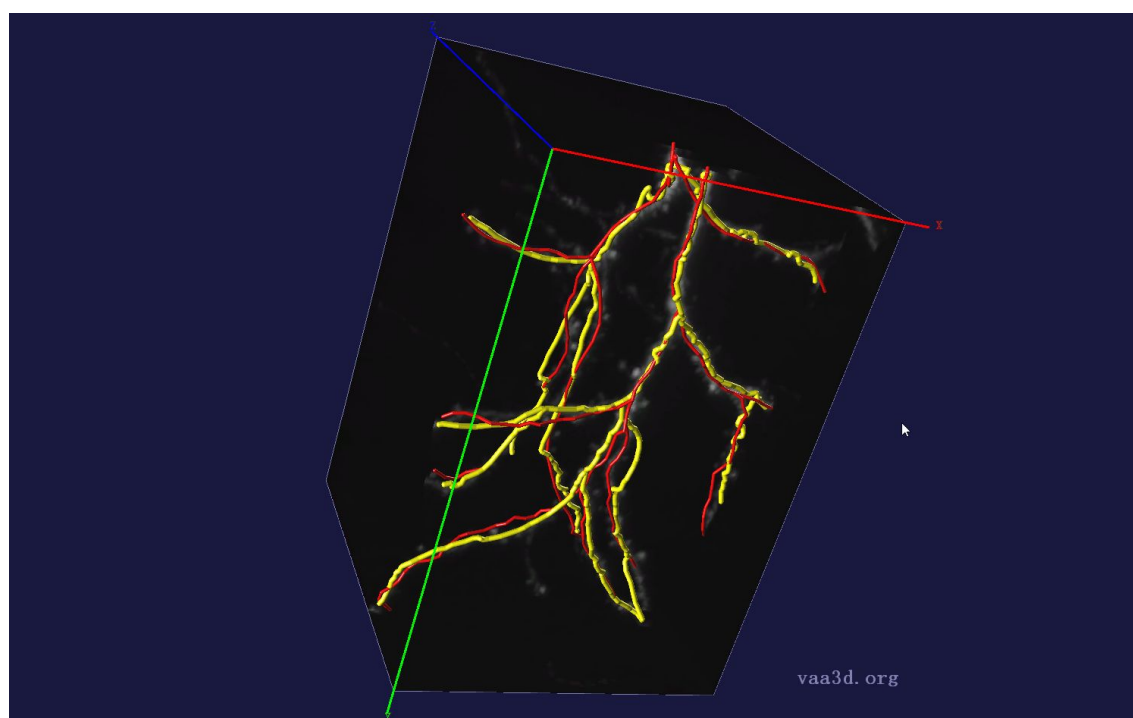


图 5.12 两种实验结果对比

从对比图中我们可以看出，大部分图像区域两次试验结果都基本贴合图像信号，但相比之下，第一次实验产生的曲线（黄色）不够平滑，多有上下抖动的情况，而第二次实验产生的曲线（红色）则足够平滑，更符合神经元的实际立体结构。另一方面，在图像前后亮信号重叠的复杂区域，第一次实验结果与实际图像信号的误差较大，而第二次实验结果更加贴合图像信号，更加准确。以上结果表明，本可视化系统手动标注产生的重建结果精确度更高，曲线更加平滑，能够更好地处理复杂结构。

### 5.3 结论

经过两次对比实验分析，可以得出以下结论：

- 1) 本可视化系统中的手动重建方法相比传统的 2D 屏幕上的手动重建方法



具有精确度更高，重建效率更快，适用于复杂结构图像的优点；

- 2) 本可视化系统所设计的智能交互方式，即半自动重建方式，极大地简化了用户重建的操作，在能够保证精确度足够高的前提下，大大减少了神经元几何形态重建所需时间与精力，能够在相同时间内获得更多不同神经元较为精确的重建结果，从而促进神经元机制等相关研究。

## 第六章 结论与展望

### 6.1 结论

本文针对神经元几何形态重建领域，结合前沿的虚拟现实技术，设计并实现了隶属于 Vaa3D 平台的脑神经图像数据在虚拟现实环境中的可视化分析与交互系统。该系统能够对多种格式不同规模的三维神经元图像数据进行虚拟现实环境中的可视化分析，并且给用户提供了方便高效的交互方式，用户可以在虚拟现实环境中实现神经元图像信号的表面 SWC 的标注和关键点 marker 的标注，最终帮助用户直观，高效，便捷地进行神经元几何形态重建的工作。

### 6.2 展望

当然，本论文设计的可视化系统仍然存在一些不足与值得继续研究的方面：

- 1) 提供给用户的 UI 界面较为混乱繁琐，缺乏简洁有效的 UI；
- 2) 使用过程中缺少有效的提示信息，用户短时间内较难理解所有的系统功能的用途和操作方法；
- 3) 系统目前只能对系统中生成的标注（SWC/marker）进行删改操作，无法对用户载入的标注文件进行删改操作；
- 4) 系统仍然可以做进一步的性能优化，以便于将来能够流畅的分析处理更大规模（TB 级别）的数据；
- 5) 由于虚拟现实技术本身的技术瓶颈，用户难以长时间的在虚拟现实环境中工作、观察，此为虚拟现实领域普遍技术难题。

## 参考文献

- 【1】. Wang C W, Lee Y C, Pradana H, et al. Ensemble Neuron Tracer for 3D Neuron Reconstruction[J]. Neuroinformatics, 2017, 15(2):1-14.
- 【2】. Peng H, Hawrylycz M, Roskams J, et al. BigNeuron: Large-Scale 3D Neuron Reconstruction from Optical Microscopy Images[J]. Neuron, 2015, 87(2):252-256.
- 【3】. Jung, B., Hoffhenke, M., and Wachsmuth, I., Virtual Assembly With Construction Kits [M]. Proceedings of 1997 ASME Design Engineering Technical Conference, September 14-17, 1997, Sacramento, DETC97/DFM-4363
- 【4】. 张茂军. 虚拟现实系统[M]. 北京: 科学出版社, 2001.9: 114-169
- 【5】. 555
- 【6】. 蔺想红, 赵吉昌, 李志强,等. 三维神经元几何形态的发育生成方法[J]. 计算机工程, 2017, 43(10):302-309.
- 【7】. Peng H, Bria A, Zhou Z, et al. Extensible visualization and analysis for multidimensional images using Vaa3D.[J]. Nature Protocols, 2014, 9(1):193.
- 【8】. Jerald J. The VR Book: Human-Centered Design for Virtual Reality[J]. 2015.
- 【9】. 冀盼, 谢懿德. VR 开发实战[M]. 电子工业出版社, 2017
- 【10】. 徐兆吉, 马君, 何仲,等. 虚拟现实:开启现实与梦想之门[M]. 人民邮电出版社, 2016.
- 【11】. Egger J, Gall M, Wallner J, et al. HTC Vive MeVisLab integration via OpenVR for medical applications[J]. Plos One, 2017, 12(3):e0173972.
- 【12】. 胡迎松, 彭利文, 池楚兵. 基于.NET的Web应用三层结构设计技术[J]. 计算机工程, 2003, 29(8):173-175.
- 【13】. 苗志宏, 马金强. 虚拟现实技术基础与应用[M]. 清华大学出版社, 2014.
- 【14】. 胡宝平. 基于 ITK 和 OpenGL 的医学图像三维表面重建系统研究[D]. 第四军医大学, 2009.
- 【15】. 尹学松, 张谦, 吴国华,等. 四种体绘制算法的分析与评价[J]. 计算机工程与应用, 2004, 40(16):97-100.
- 【16】. 李冠峰, 黄毓瑜, 杨光. 体可视化的快速光线投射算法[J]. 图学学报, 2000, 21(3):97-102.
- 【17】. Peng H, Bria A, Zhou Z, et al. Extensible visualization and analysis for multidimensional images using Vaa3D.[J]. Nature Protocols, 2014, 9(1):193.
- 【18】. Peng H, Tang J, Xiao H, et al. Virtual finger boosts three-dimensional imaging and microsurgery as well as terabyte volume image visualization and analysis[J]. Nature Communications, 2014, 5(5):4342.
- 【19】. Sethian J A. Fast Marching Methods[J]. Siam Review, 2015, 41(2):199-235.

## 作者在攻读硕士学位期间公开发表的论文

## 作者在攻读硕士学位期间所作的项目

## 致 谢

本文是在导师王宜敏老师的悉心指导下完成的。承蒙王老师的亲切关怀和精心指导，虽然有繁忙的工作，但仍抽出时间给予我学术上的指导和帮助，特别是给我提供了良好的学习环境，使我从中获益不浅。从毕业设计的开题阶段，到中期的实验阶段，再到论文的写作、修改，最终到论文定稿，王老师给了我耐心的指导和无私的帮助。王老师在我算法编写遇到困难时对我的耐心指导，对我论文初稿细致入微的修订与建设性的修改建议都让我印象深刻，感激不尽。衷心的祝愿王老师能够永远身体健康，事业顺利！

同时我还要感谢在我学习期间给我极大关心和支持的各位老师以及关心我的同学和朋友。同时，感谢所有任课老师和所有同学在这两年来给自己的指导和帮助，是他们教会了我专业知识，教会了我如何学习，教会了我如何做人。正是由于他们，我才能在各方面取得显著的进步，在此向他们表示我由衷的谢意，并祝所有的老师培养出越来越多的优秀人才，桃李满天下！

写作毕业论文是一次再系统学习的过程，毕业论文的完成，同样也意味着新的学习生活的开始。我会在之后人生道路上铭记这一经历，不断前行，不断努力！

感谢各位专家的批评指导！