

题 目 基于VTK的三维可视化平台开发

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

计算机科学与工程院（系）计算机科学与技术专业

学 号 09015322

学生姓名 贺建安

指导教师 唐慧

起止日期 2019年2月 至 2019年5月

设计地点 东南大学九龙湖校区

摘 要

传递函数是直接体绘制的核心，它决定着最终的绘制效果。传统的以用户为中心的交互式传递函数设计是一个不断试验的过程，存在一定的盲目性，且难以设计出以不同样式来绘制不同组织结构的传递函数。本文基于VTK与Qt设计了一个三维可视化平台，

在传递函数设计方式上，平台既提供了友好的用户交互式设计方式，也提供了基于体数据分析自动生成传递函数的设计方式。两种方式相互融合，使得交互复杂度与数据分析计算量都大大降低，且所有的交互设置结果与数据分析结果都可以实时地在切片图或体绘制图上得到反馈。

在可视化边界的传递函数生成过程中，本文使用非结构化数据存储阈值过滤后的体数据，采用K-Means算法，结合边界的梯度幅值的属性，生成基于灰度值与梯度幅值的二维不透明传递函数，用于绘制用户感兴趣区域的边界。

此外，通过可视化传递函数，用户可以对传递函数进行微调，调节结果实时反馈在体绘制图上。也支持多个感兴趣区域、多个感兴趣区域边界以及多个体数据集的叠加绘制。

关键词：直接体绘制，传递函数，边界可视化

Abstract

In this thesis, the rheological behavior of semi-solid ZA12 alloy was investigated using a specially designed high temperature Couette rheometer.

The evolution of shear stress with time and the hysteresis loops of semi-solid ZA12 alloy were measured and analyzed. The results show that semi-solid ZA12 alloy possesses the thixotropic property, which varies with solid fraction and shear rate. In addition, the semi-solid ZA12 alloy slurry exhibits different rheological behaviors under steady state and transient state conditions. In case of steady state, the apparent viscosity of semi-solid ZA12 alloy decreases with the increase of shear rate, showing the pseudo-plastic rheological behavior. However, under the transient state condition, it presents the dilatant rheological behavior, i.e. the apparent viscosity increases as shear rate increases.

Finally, based on the transient state experimental results and rheology theory, a dynamic rheological model of semi-solid ZA12 alloy was developed, which could be applicable to practical semi-solid processes.

KEY WORDS: semi-solid, ZA12 alloy, thixotropic behavior, rheological behavior, apparent viscosity, rheological model

目录

[摘 要 2](#_Toc8243001)

[Abstract 3](#_Toc8243002)

[第一章 绪论 1](#_Toc8243003)

[1.1 研究背景与意义 1](#_Toc8243004)

[1.2 三维可视化的研究现状与发展趋势 2](#_Toc8243005)

[1.3 论文的主要内容与组织结构 3](#_Toc8243006)

[第二章 相关知识 4](#_Toc8243007)

[2.1 光线投射算法 4](#_Toc8243008)

[2.1.1 光照模型 4](#_Toc8243009)

[2.1.2 基本原理 6](#_Toc8243010)

[2.1.3 算法改进 9](#_Toc8243011)

[2.2 传递函数定义 9](#_Toc8243012)

[参考文献： 10](#_Toc8243013)

# 第一章 绪论

## 1.1 研究背景与意义

科学计算可视化(Visualization in Scientific Computing)是一个于20世纪80年代被提出并迅速发展起来的一个研究领域。它指的是运用计算机图形学和图像处理技术，将科学计算过程中及计算结果的数据转化为图形或图像在屏幕上显示出来并能够进行交互处理的理论、方法和技术[1]。借助科学计算可视化，人们可以更加直观地理解计算过程中的数据变化，加以交互实现对计算过程的引导与控制，直观的图像或图形可视化也能帮助人们更好地理解数据中所蕴含的抽象信息。

科学计算可视化的核心是三维空间数据场的可视化，又称三维体数据可视化，是一种将三维体数据在二维平面上进行投影的技术，它被广泛应用于医疗、地质勘探、气象分析等领域。实现三维体数据可视化的算法可分为两类，一类是基于中间几何图元的面绘制方法，另一类是直接基于三维体数据的体绘制方法。

面绘制即是以等值面的形式来绘制三维模型，其优点是计算量较小，但无法反映整个原始数据场的全貌及细节，且该方法始终存在等值面分类的歧义性问题，因此当数据场较为复杂时，往往会生成不存在的伪表面或在现实面上产生空洞[3]。

体绘制是直接为三维体数据场中的体素分配颜色与不透明度等光学性质进行绘制，而不需要进行中间图元的转换，因此也被称为直接体绘制(Direct Volume Render, DVR)。直接体绘制不仅能够展现出表面轮廓，也能清晰地还原出立体细节，不会造成数据的丢失。因此，在例如医学图像三维重建等对可视化效果要求较高的应用领域，直接体绘制往往是首选的可视化技术。

传递函数(Transfer Function, TF)是直接体绘制的基础，它的本质作用是将颜色与不透明度等光学性质分配给体素数据，从而使数据可视。良好的传递函数可以揭示数据中的重要结构，而不会用不重要的区域来模糊它们[6]。因此，传递函数的设计，决定着直接体绘制的最终绘制效果。然而，要找到一个能够满足用户预期绘制效果的传递函数往往是一个挑战，原因主要体现在以下四个方面。其一，传递函数的优劣无法通过量化的指标来评定，人们往往只能通过观察直接体绘制最终的绘制效果，来间接地判断传递函数的设计是否合适，存在一定的主观性；其二，传递函数的调节不受原数据集的约束或指导，这迫使用户进入一个不断实验的交互模式，在这种模式下，只有通过观察由于增量调整而导致的绘制效果的变化，才能探索传递函数域[6]；其三，传递函数的调节自由度很大，从三维体数据场中的各种标量值到各种光学性质存在多个映射函数，映射关系与绘制效果之间也没有绝对的规律可寻，传递函数的细微变化可能会导致绘制效果的巨大差异，调节具有盲目性；其四，传递函数往往是作用于整个三维体数据场，它的定义域并不包含空间位置，因此当用户从空间位置上来确定感兴趣区域时，是很难通过设计传递函数来实现区域分割的。因此，如何设计一个好的传递函数，已经被列为三维体数据可视化研究中的十大难题之一[7]。传递函数设计有着重要的研究意义与急迫的研究需求，这也是本文的研究重点。

## 1.2 三维可视化的研究现状与发展趋势

作为科学计算可视化的关键问题，三维体数据可视化一直是一个研究热点。目前较为成熟的直接体绘制算法有：光线投射算法(ray-casting)、抛雪球算法(Splatting)、错切-形变算法(Shear-warp)与3D纹理映射(3D Texture Mapping)，其中绘制图品质最高的是光线投射算法[4][5]。光线投射算法是基于物理光线模型，把每个体素看作是可以发射、反射和吸收光线的粒子，依据体素的介质特征得到它们的颜色与透明度，并沿着视线观察方向积分，最后在绘制窗口形成具有半透明效果的图像[5]。且该算法可并行计算，随着图像处理器的并行处理能力的提升，使用该算法进行三维体绘制时，用户可实时地对绘制图进行旋转、缩放等交互行为。

虽然直接体绘制算法已经在绘制效果与交互速率上满足可视化的要求，但是作为直接体绘制的基础，传递函数的设计依然是一个瓶颈，限制着体绘制的发展。传递函数的设计方法可分为四大类：手动调节法、图像中心法、数据中心法和对象中心法[8]。

(1)手动调节法是最原始也是最基础的传递函数设计方法。该方法在设计过程中，需要用户不断地去调节传递函数的参数，同时增量地观察绘制图的变化，直到绘制效果满意为止。该设计方法给予用户的自由度最大，所需的计算量也最小，但是需要耗费用户大量的时间与精力去调节，效率最为低下。但大部分提供三维可视化功能的软件中都会提供这种设计方式，以保证用户对体绘制效果的自由控制。

(2)图像中心法的关注对象是体绘制图像。该方法让用户评估一系列具有不同绘制效果的体绘制图，通过用户的评估来改变传递函数，再生成新的体绘制图让用户进行评估，直到得到用户最满意的绘制效果。The Design Gallery是这种方法的代表，它基于自动化分析与已绘制图像，为所有可能的传递函数域创建了一个直观的接口供用户选择[9]。该方法的关键是用户的评估以及根据评估修改传递函数的方式[8],其本质是以最小化信息熵的形式构建用户输入的反馈过程，可归结为优化问题[10]。

(3)数据中心法是关注的是三维体数据本身的数据特征。传递函数的定义域一般是灰度值，梯度幅值等三维体数据场中的局部数据属性，由这些局部数据计算得到的全局数据属性，如等值面、边界面和拓扑结构等，往往标识着三维体数据中某一特定的结构，因此可利用这些全局特征来指导传递函数的设计。Kindlmann等人提出的一种半自动生成传递函数的方法[11]，就是利用原数据集中的灰度值，梯度幅值以及二阶梯度幅值组成的直方体(Histogram Volume)来定位边界，从而设计出能够绘制物质边界的传递函数。

(4)对象中心法是先对三维体数据场中的体素进行分类，再对不同的类别分配不同的颜色与不透明度等光学性质。该方法的关键是分类。Tzeng等人提出一种类似于K-Means聚类的无监督机器学习算法ISODATA[12]，该算法根据空间距离对三维体数据进行分类，并提供交互方式使用户可以直接对分类结果进行修改。近年来，随着机器学习的兴起，越来越多的基于机器学习的对象中心设计方法被提出。Tzeng等人基于最简单的BP网络，通过用户选取的感兴趣区域采样点及不感兴趣区域采样点来训练网络，实现分类[13]。HC Cheng等人使用卷积神经网络从三维体数据中自动地提取高层次的分类特征，指导传递函数的生成[14]。本质上，该设计方法可以看作是数据中心法的扩展，因为都是根据体数据自身的特征来指导传递函数的设计。

综上所述，传递函数的设计方法研究一直是三维体数据可视化的一个研究热点，更智能、更友好、更高效的设计方式不断被提出。而随着三维体数据的数据量的增加，数据结构复杂度的提升，以及实时可视化的需求，传递函数设计将会有以下几个研究方向：(1)基于压缩的原数据进行传递函数设计。超高分辨率的体数据集常以压缩格式保存，基于压缩数据设计传递函数，可大大节省存储空间与绘制时间；(2)多维传递函数的设计与简化。更高维的传递函数可以表示更多的特征，从而能更精细地进行绘制，但设计的复杂度与计算量也随之提升，因此在传递函数维度的选取上需要做到最优；(3)融入用户视角。三维体绘制是根据用户视角展现一部分数据场，将视角融入传递函数的设计，可以在非交互式的三维体绘制中通过最佳视角，展现重要的数据特征。

1.3 论文的主要内容与组织结构

第二章 体绘制算法与传递函数

三维体数据场是通过对三维连续空间进行等距采样，得到的一种由均匀规则网格组成的结构化数据，每个网格顶点为一个采样点，称为体素(Voxel)，是三维体数据场的基本元素。三维体数据场的函数值分布于每个体素中，即数据场中位于顶点处的函数值为 (如图2.1所示)。通过计算机断层扫描(CT)或磁共振成像(MRI)得到的二维图像序列组合而成的数据就属于这一类型。

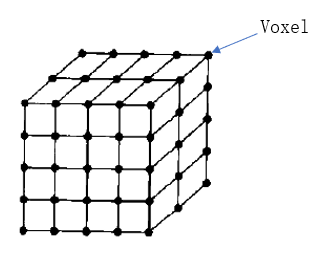


图2.1 三维体数据场示意图。

直接体绘制算法(以下简称体绘制算法)以其高质量的成像品质，在三维体数据可视化领域中被广泛使用。体绘制算法的实质是将离散的三维体数据场按照一定规则，转换为二维离散信号，即每个像素点的RGB颜色值，存储到图形显示设备的帧缓冲中。本章将详细介绍光线投射体绘制算法，以及决定其绘制效果的传递函数的定义。

## 2.1 光线投射算法

如前所述，体绘制算法是将三维空间的离散数据按一定规则转化为二维离散信号，转化过程中，需要对三维体数据进行重采样，然后计算每个采样点对二维图像像素的贡献，并将将所有贡献进行合成。光线投射体绘制算法是M.Levoy等人于1988年提出，是一种基于图像空间扫描，应用光学模型进行图像合成的体绘制方法[2]。本节将首先介绍常用的几种光学模型，在此基础上介绍光线投射算法的基本原理，最后简要介绍光学投射算法的几种改进方法。

### 2.1.1 光学模型

三维体数据场中的数据原本是不具备颜色属性的，它们的颜色值通过传递函数进行映射的。体绘制中，计算每个采样点对二维图像像素的贡献时，就是在计算每个像素的光强度值I(Intensity)。在灰度图像中，I表示的就是灰度值，I=0表示黑色，I=1表示白色。在彩色图像中，红、绿、蓝三个分量都有各自的I值。

为了计算各采样点光强对二维图像像素的贡献，我们需要给出光学模型，用来描述三维体数据场是如何产生、反射、散射以及吸收光线的。

Nelson Max在“Optical models for direct volume rendering”一文中提出假设：“空间中连续分布的三维数据场中充满着小粒子，由于这些小粒子的发光、吸收、反射等功能使得光线通过三维数据场时发生了变化”。基于这一假设形成了几种不同的光学模型。

1. 光线吸收模型

假设三维空间中的小粒子可完全吸收所接收的光线，而无发射或反射光线的能力。基于这一假设就构成了一个光线吸收模型，这是最简单的一种光学模型。光线吸收模型中，光线通过三维数据场时，光强的表达式为：

|  |  |  |
| --- | --- | --- |
|  |  | (2.1) |

式中，为光线沿投射方向的行经长度，表示入射点的初始光强，表示光线与入射点距离时的光强。是光强的衰减系数，与三维空间中的粒子密度，以及每个粒子的投影面积成正比。令

|  |  |  |
| --- | --- | --- |
|  |  | (2.2) |

表示光线穿越体数据场距离后的强度衰减率，也称为透明度。体绘制中，常使用来表示不透明度：

|  |  |  |
| --- | --- | --- |
|  |  | (2.3) |

体绘制中，通过不透明度传递函数来指定三维体数据场中数据的不透明度。若将设为1，则表示该数据代表的物质完全不透明，穿过它的光线会被完全吸收，因此在绘制时，位于在该物质后面的物质会被遮挡而不可见。反之，若将设为0，则表示该数据代表的物质完全透明，绘制时该物质不可见。

1. 光线发射模型

在例如火焰、高温气体等物质的可视化中，可以假设三维空间中的小粒子是完全透明的，且自身可以发出光线，即小粒子只能发射光线，无法吸收光线。基于这一假设就构成了光线发射模型。光线发射模型中，光线通过三维数据场时，光强的表达式为：

|  |  |  |
| --- | --- | --- |
|  |  | (2.4) |

式中，表示光强的增加率，与三维空间中的粒子密度，每个粒子的投影面积，以及每个粒子所发出的光强C成正比。

1. 光线吸收与发射模型

将光线吸收模型与光线发射模型组合在一起，可构成光线吸收与发射模型，即三维数据场中的小粒子既可以吸收光线，也可以发射光线，该模型可以更加真实地反映出光线在充满粒子的三维空间中的变化。光线吸收与发射模型中，光强的表达式为：

|  |  |  |
| --- | --- | --- |
|  |  | (2.5) |

式中，表示不透明度。该式表示初始光强为的光线，通过所赋颜色值为，透明度为的物质后的光强。该模型可有效应用于通过CT或者MRI测量得到的三维医学数据的可视化中。

### 2.1.2 基本原理

光线投射体绘制算法的流程图如下图所示。

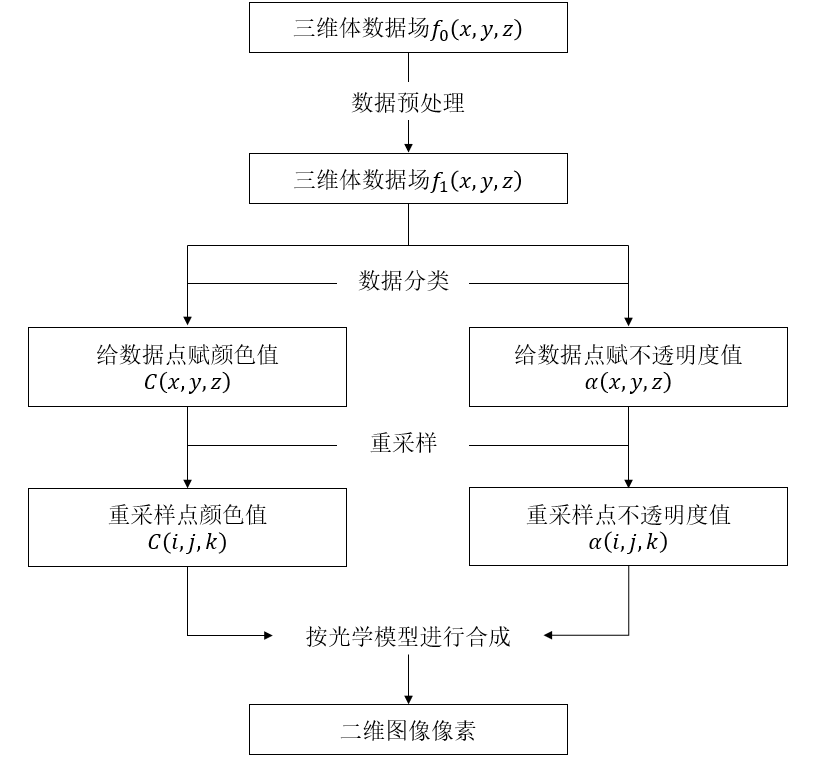


图2.2 光线投影算法流程图。

数据预处理包括数据格式转换，冗余数据去除等操作。数据分类是根据三维体数据场中的数据特征，将体数据划分为若干个类，每一类赋予不同的颜色值和不透明度值，以求在可视化时表现出多种物质的不同分布，或同一物质的不同属性，具体的实现由传递函数来完成。接下来两个步骤是重采样与图像合成。

1. 重采样

使用光线投射算法进行可视化时，是从屏幕上的每一个像素点，沿着视线方向引出一条射线，这条射线穿过三维体数据场。由于三维体数据场中的原始采样点是离散且均匀地分布的在每个网格的顶点，穿越体数据场的射线不一定与原始采样点相交，因此要对三维体数据场进行重采样，以得到该射线与三维体数据场的交点。重采样时，沿着射线选择K各等距的重采样点，某个重采样点的颜色值与不透明度值可通过插值计算得到。下图是重采样的二维形式图解。

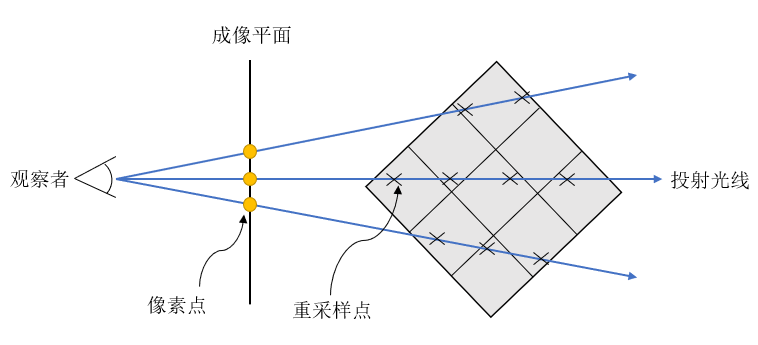


图2.3 重采样图解。

三维体绘制中较为常用常用的插值方法有最近邻插值与三次线性插值。最近邻插值是直接将与重采样点最近的体数据点的颜色值及不透明度值赋给该采样点，该插值方法计算量较小，但可能造成数据值的不连续，从而在绘制图上出现锯齿；三次线性插值是首先定位到与重采样点距离最近的8个体数据点，然后根据距离对8个体数据点的值进行加权平均，得到重采样点的值。下图是使用三次线性插值进行重采样。

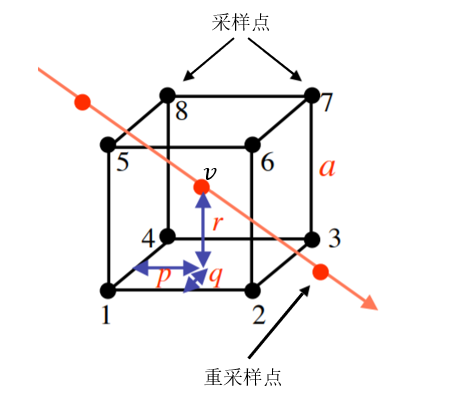


图2.4 使用三次线性插值进行重采样。

对应的插值公式为：

|  |  |  |
| --- | --- | --- |
|  |  | (2.6) |

式中，，，分别表示第个采样点与重采样点在，，轴上的距离，例如，。

1. 图像合成

通过插值计算得到一条射线上所有重采样点的颜色值与不透明度值之后，还需要将这些值由后向前或者由前向后进行合成，合成方法遵从光线吸收与发射模型。设第个重采样点的颜色值为，不透明度值为；进入第个重采样点的光线的颜色值为，不透明度值为；从第个重采样点出来的光线的颜色值为，不透明度值为。

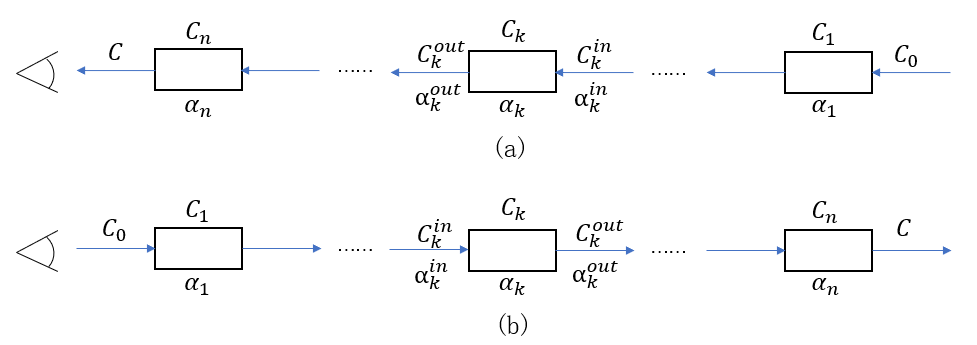


图2.5 图像合成示意图。(a)为由后向前合成，(b)为由前向后合成。

由后向前图像合成是沿着与射线相反的方向将各重采样点的颜色值与不透明度值进行合成，如2.5(a)所示。则依照式(2.5)可以得到合成公式：

|  |  |  |
| --- | --- | --- |
|  |  | (2.7) |

设初始颜色值为，第个重采样点的透明度，将式(2.7)应用于所有重采样点后，可以得到最终合成的颜色值为为：

|  |  |  |
| --- | --- | --- |
|  |  | (2.8) |

由前向后图像合成是沿着与射线方向将各重采样点的颜色值与不透明度值进行合成，如2.5(b)所示。合成公式如下：

|  |  |  |
| --- | --- | --- |
|  |  | (2.9) |

使用式(2.9)沿射线方向逐重采样点计算最终合成的颜色值时，累积不透明度值是不断增加的，当趋近于1时，说明引出该射线的像素点已经几乎完全不透明了，则当前重采样点之后的点就不会再对该像素点的颜色值有所贡献了，因此可以不用计算了。相比于由后向前图像合成法，由前向后合成法可以省去无效的计算，速度较快，因此在体绘制算法中的应用更广。

## 2.2 传递函数

如前所述，体绘制算法中的一个主要步骤就是根据三维体数据场中的数据属性进行分类，并为不同类别的数据分配不同的颜色与不透明度等光学属性。数据分类与光学属性分配是后续重采样以及图像合成的基础，决定着最终的绘制效果。从三维体数据场的数据属性到光学属性的映射，就是传递函数，可以表示为：

|  |  |  |
| --- | --- | --- |
|  |  | (2.10) |

式中，表示三维体数据场的数据属性，作为传递函数的定义域；表示用于可视化的光学属性，作为传递函数的值域；表示从数据属性到光学属性的映射规则。传递函数的设计过程就是根据可视化需求，选择合适的数据属性与光学属性，并建立起它们之间的映射规则[8]。本节将详细介绍传递函数的这三个部分。

### 2.2.1 数据属性

数据属性是三维体数据场自身的性质，是传递函数的定义域。根据传递函数定义域的维度，可以将传递函数分为一维传递函数与多维传递函数。数据属性包括局部属性与全局属性。局部属性可以是单个采样点的属性，如标量值、空间位置，也可以是几个相邻采样点共同计算得到的属性，如梯度、曲率；全局属性是指由局部属性计算得到的数据属性，如拓扑结构。下面介绍标量值、梯度以及二阶导数这三个常用的局部数据属性。

(1) 标量值

标量值是指三维体数据场中每个体素的数据值，是最简单、最常用的数据属性。标量值一般是通过仪器测量或者数值计算直接得到的数值，能够揭示数据所代表的物质的最本质特征。例如，通过读取人体CT图像序列得到的三维体数据中，标量值是图像的灰度值，而灰度值是各组织结构的CT值线性变化而来。不同的组织结构的CT值往往不同，因此，通过该三维体数据场中的标量值就可以直观地初步区分出人体的不同组织结构。只基于标量值的传递函数没有利用数据之间的特征，因此只能处理数据分布较为规整的简单三维体数据场。

(2) 梯度

通常来说，三维体数据场中，同一种物质内部的标量值很相似，而不同物质的标量值差异较大。因此，在不同物质的交界处，标量值的变化会比较剧烈。可以使用梯度来反映标量值的变化剧烈程度。对于离散的三维体数据场，最常用的梯度计算方法是中心差分法：

|  |  |  |
| --- | --- | --- |
|  |  | (2.8) |

每个体素的梯度都有三个分量，每一个轴相邻两个体素的标量值的差值平均，即是沿该轴的梯度分量。梯度幅值是三个分量的均方根：

|  |  |  |
| --- | --- | --- |
|  |  | (2.9) |

体绘制中常使用梯度幅值作为传递函数定义域来提取物质的边界面。为了保证梯度计算的准确性与连续性，线性分离法[16]以及四维线性回归法[17]也是常用的梯度计算方法。程序实现时，常使用梯度算子来进行梯度计算。

(3) 二阶导数

沿梯度方向的二阶导数能够反映出梯度幅值的变化。理想情况下，物质边界面的梯度幅值处于极值，沿梯度方向的二阶导数为0。使用二阶导数，或者二阶导数与梯度幅值共同作为传递函数定义域时，物质边界的检测效果会比只使用梯度幅值要好[11]。梯度方向的二阶导数计算有三种方法[11]：

1. 利用梯度幅值进行计算:

|  |  |  |
| --- | --- | --- |
|  |  | (2.10) |

1. 利用Hessian矩阵进行计算：

|  |  |  |
| --- | --- | --- |
|  |  | (2.11) |

其中为Hessian矩阵，通过对求二阶偏导得出[19]。该方法计算较为准确但计算量较大。利用Hessian矩阵的特征值还可以对三维体数据场中的局部结构进行识别[20]。

1. 利用Laplacian公式近似计算：

|  |  |  |
| --- | --- | --- |
|  |  | (2.12) |

该方法计算量较小，但是对噪声敏感。具体实现时常通过Laplacian算子来计算。

### 2.2.2 光学属性

光学属性是是传递函数的值域，是用于可视化的属性。最常用的光学属性有颜色值与不透明度值。颜色通常采用的是RGB模型或者HSV模型，将数据场中不同的物质赋予不同的颜色值，可以在视觉上进行区分；不透明度值位于0到1之间，用来控制物质的可见度，对感兴趣物质设置高不透明度以突出，对不感兴趣物质设置低不透明度以隐藏。

除了颜色与不透明度，其他光学属性，如折射率、反射率、阴影参数等也可以作为传递函数的值域。例如，Lum等人采用光学属性中的阴影参数作为传递函数值域，设计出光照传递函数(lighting transfer function)[18],该传递函数通过调节物质边界面的阴影，能更突出边界，而且能表现出物质的厚度。

### 2.2.3 映射规则

传递函数的映射规则是指数据属性映射到光学属性的规则。当传递函数定义域只有一维时，常用的映射规则有梯形函数、矩形函数、三角函数和斜坡函数，示意图如下：

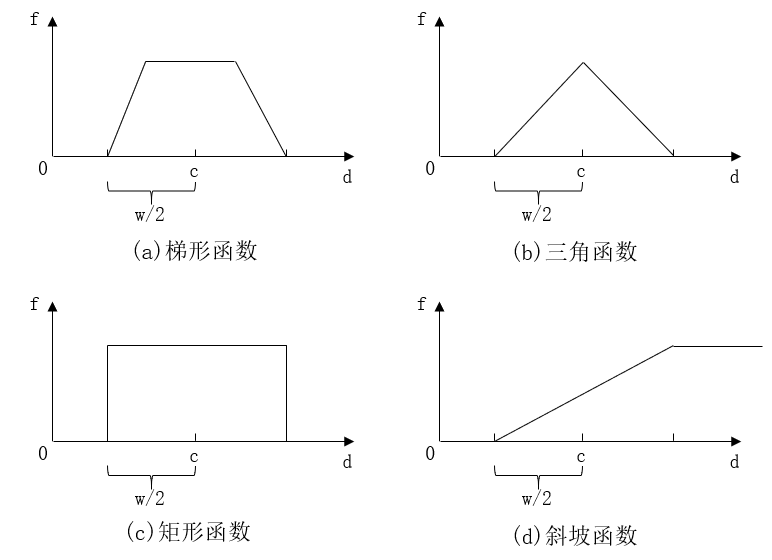


图2.6 传递函数映射规则。

式中，c表示感兴趣区域的中心，w表示区域范围。对于多维传递函数，自变量数目大于1，映射规则的设计自由度也更大。Kniss等人提出高斯传递函数(Gaussian Transfer Function, GTF)来表示多维传递函数的映射规则：

|  |  |  |
| --- | --- | --- |
|  |  | (2.13) |

式中，是多个自变量所组成的向量，和分别是中心向量和线性转换矩阵，用来控制映射以为中心进行旋转或者放缩。

第三章 三维可视化平台概述

## 3.1 平台开发环境

本文基于Qt5以及VTK8.2.0实现了一个三维可视化平台。其中VTK主要负责数据读取、处理以及可视化，Qt主要负责平台界面设计以及用户交互(其中对可视化图像的旋转放缩等交互由VTK实现)。

Qt是C++图形界面应用程序的开发框架，以其可扩展性与跨平台性而被广泛使用。本平台的可视化界面是基于Qt Widgets Application进行搭建，通过Qt自带控件以及自定义控件为传递函数的设计提供了友好的用户接口，使用Qt的信号与槽机制处理用户事件。

VTK全称是The Visualization Toolkit，由Kitware公司所开发，是一个开源的，面向对象的软件开发工具包，可用于计算机图形学、可视化以及图像处理。VTK的两个主要组成部分是可视化管道(The visualization pipeline)与绘制引擎(The rendering engine)。

可视化管道可用来创建与分析数据。VTK支持多种结构类型的数据对象，如图3.1所示。数据的分析处理是通过Filter来完成，一个Filter具备某一特定的数据处理功能，例如类型转换、阈值分割等；多个Filter进行组合可构成可视化管道，数据以流的形式在管道中传递，如图3.2所示。绘制引擎接收可视化管道的输出，负责可视化过程，它包含各种可视化相关的组件，如vtkCamera可用来调节相机属性，vtkLight可用来控制光照属性。

(a) Image Data 
(vtklmageData) 
(c) Structured Grid 
(vtkStructuredGrid) 
(e) Polygonal Data 
(vtkPolyData) 
(b) Rectilinear Grid 
(vtkRect11inearGrid) 
(d) Unstructured Points 
(use vtkPotyData) 
(f) Unstructured 
(vtkUnstructuredGrid) 

图3.1 VTK所支持的数据类型。

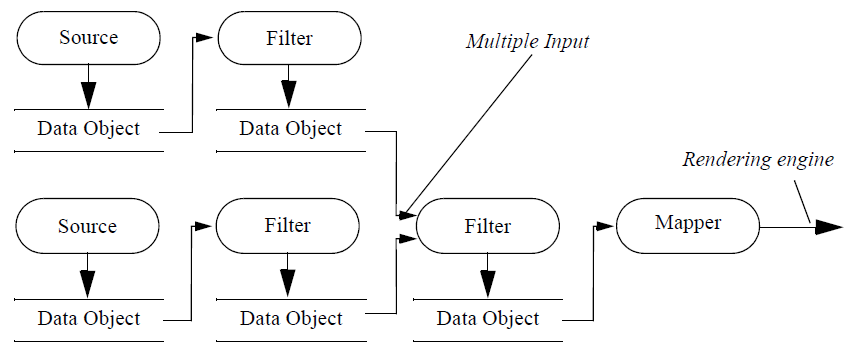


图3.2 VTK可视化管道示意图。

VTK的一个重要机制是lazy evaluation，即只有当数据真正被需要时，计算才会开始。这种机制能带来两个好处，一是推迟计算，减少不必要数据处理与存储空间；二是当管道的输入改变时，不需要显式地刷新管道中的各个Filter，只需刷新最后一个Filter就可以得到新的输出结果，从而减少代码量。

本平台使用VTK的C++接口，主要使用的是vtkImageData这种规则的数据结构来表示三维体数据场，通过调用各种Filter以及自定义的数据处理算法来搭建可视化管道，并与三维体绘制相关的绘制组件相连，实现三维可视化流程。

## 3.2 平台概述

### 3.2.1 平台各模块概述

本平台一共四个模块，分别是：

(1) 数据读取模块

该模块是平台的入口，用户通过该模块读入DICOM图形序列，或者NIFTI格式的数据，读取一些图像的基本信息，然后建立起三维体数据场，并传递给三维体绘制模块进行可视化，或者传递给数据分析模块进行分析。

(2) 三维体绘制模块

该模块基于VTK搭建的三维可视化流程，该流程的输入是三维体数据，输出是屏幕上的二维绘制图像，并可接收传递函数来改变可视化的效果。

(3) 数据分析模块

该模块的作用是分析传入的三维体数据的特征，得到的分析结果既可以用来指导用户进行交互式地传递函数设计，也可以用来自动生成传递函数。

(4) 传递函数设计

该模块是本平台的核心，共提供两种传递函数设计方法，一种是通过友好的用户接口实现实时的交互式设计，另一种是以对象为中心的自动化设计，通常是两种方法相结合，以半自动化的方式生成能够提供良好绘制效果的传递函数。

各模块之间的关系如图3.1所示

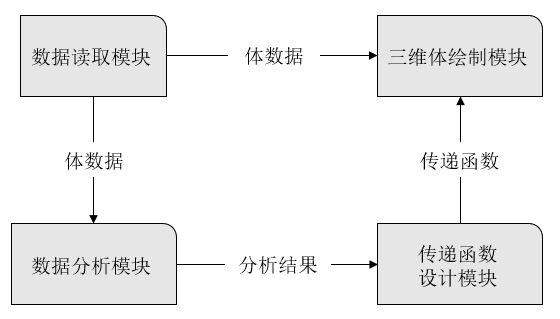


图3.1 本平台各模块关系示意图。

### 3.2.2 平台主界面概述

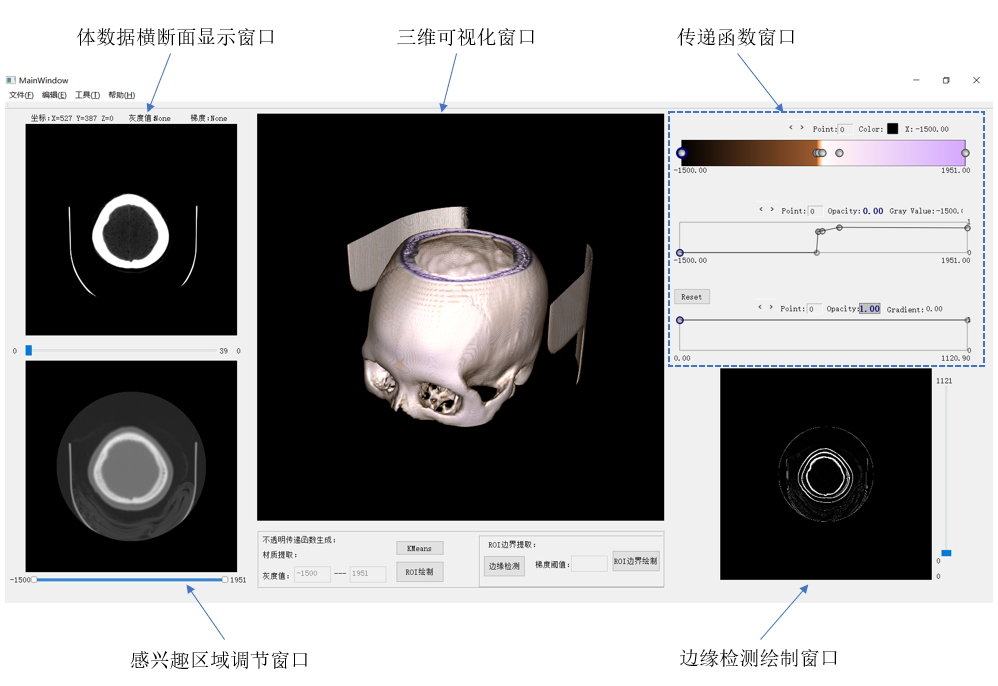


图3.2 本平台主界面。

图3.2为本平台的主界面，主要包含三个窗口：

(1) 三维体绘制窗口

最中心的大窗口，三维体绘制模块的可视化输出，用户可在本窗口使用鼠标对绘制图进行选择与放缩。

(2) 传递函数窗口

传递函数窗口可视化了3个一维的传递函数：颜色传递函数、标量-不透明度传递函数以及梯度-不透明传递函数，是传递函数设计模块的基础。传递函数窗口与三维体绘制窗口动态绑定，即传递函数窗口的传递函数发生变化时，三维体绘制图会实时地发生相应的变化。传递函数的变化既可以是用户手动修改，也可以是数据分析模块自动修改。

(3)切片显示窗口

体数据横断面显示窗口、感兴趣区域调节窗口以及边缘检测绘制窗口以二维切片形式分别显示的原始体数据值、灰度映射后的感兴趣区域以及体数据的边缘。三个窗口可通过鼠标滚轮或者滑条同步切换切片，这三个窗口是数据分析模块与传递函数设计模块的纽带。

用户可以在体数据横断面显示窗口点击查看三维体数据场中各点的标量值与梯度值，指导感兴趣区域的选择以及传递函数的交互式设计；感兴趣区域调节窗口用来选择感兴趣区域，数据分析模块是基于用户选择的感兴趣区域来进行计算；边缘检测绘制窗口显示体数据的边缘，可通过梯度滑条调节边缘梯度阈值，该窗口可指导用户调节梯度-不透明度传递函数，也可以进行数据分析自动调节该传递函数。

### 3.2.3 平台流程图概述

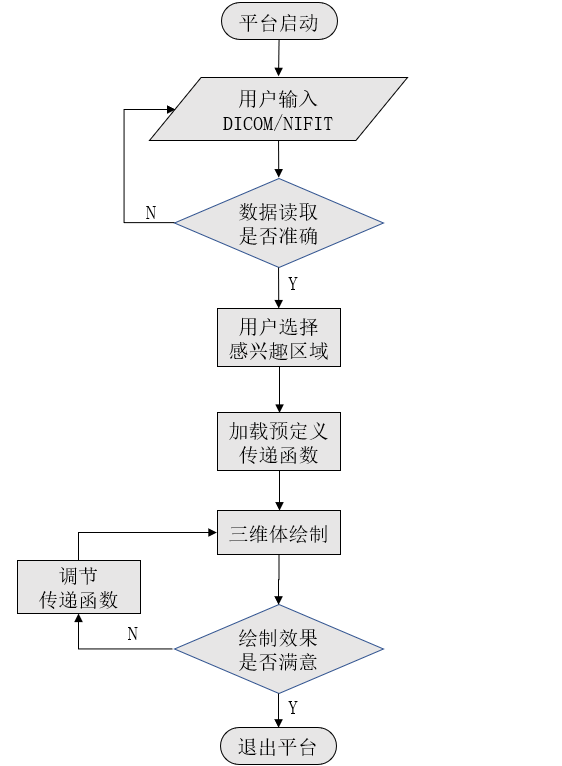


图3.3 平台的基本流程。

图3.3是本平台基本的系统流程图，用户通过点击菜单读入文件。当读入文件格式正确，三维体绘制模块会开始建立体绘制管道。然后由用户基于标量值选择一个感兴趣区域，以及预先定义的传递函数样式，进行三维体数据的初始绘制。由用户观测绘制效果，若不满意，则进入传递函数调节阶段，可直接在三个传递函数可视化图形上进行交互式修改，也可以通过数据分析自动修改；若效果满意，则进行下一步操作，如保存当前传递函数、绘制当前体数据其他结构或添加新的体数据，后两个功能是由体绘制模块提供的增量绘制功能实现，将在4.2.2小节详细介绍。

除了最核心的三维体绘制以及传递函数设计功能，本平台还提供了其他许多小功能，帮助提高用户在设计传递函时的用户体验，例如通过修改三维可视化窗口的背景颜色；显示绘制对象的边界框；快速旋转体绘制图到指定视角；键盘交互等等。总而言之，该三维可视化平台旨在为用户提供一个更加友好、更加智能的传递函数设计环境。

## 3.3 面向对象的平台设计

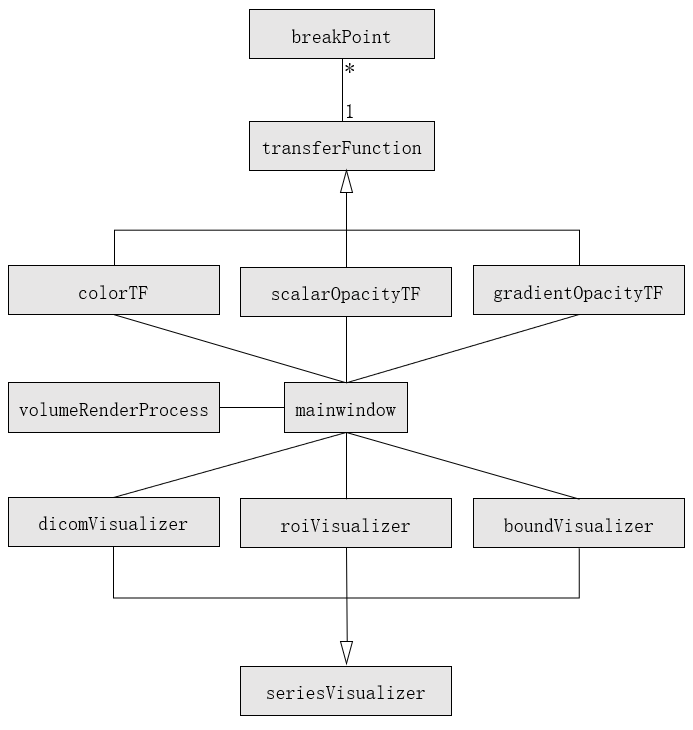


图3.4 平台的类图。

本平台采用面向对象的设计方法，类图如图3.4所示。其中，mianwindow类是平台主界面对应的类；volumeRenderProcess是三维体绘制流程对应的类；colorTF、scalarOpacityTF与gradientOpacityTF分别是颜色传递函数、标量-不透明度传递函数以及梯度-不透明传递函数对应的类，这三个类继承自模板类transferFunction，该模板类包含多个breakPoint模板类的实例；dicomVisualizer、roiVisualizer与boundVisualizer分别是三个切片窗口对应的类，数据分析算法也包含在这三个类中，这三个类继承自抽象类seriesVisualizer。

本平台的功能较多，各个类对象之间的通信也较为复杂，因此采用类似于星型拓扑结构的通信连接方式，即不同类型的类对象的通信都要通过mainwindow类对象来传递。例如，用户修改感兴趣区域，来改变标量-不透明度传递函，进而改变体绘制效果，这样一个事件的顺序图如图3.5所示，mainwindow类对象是所有Message的发送者或者接收者。这样一种通信机制虽然会稍微减低通信效率，但是有助于对所有信号的管控，避免通信紊乱，减少软件错误。

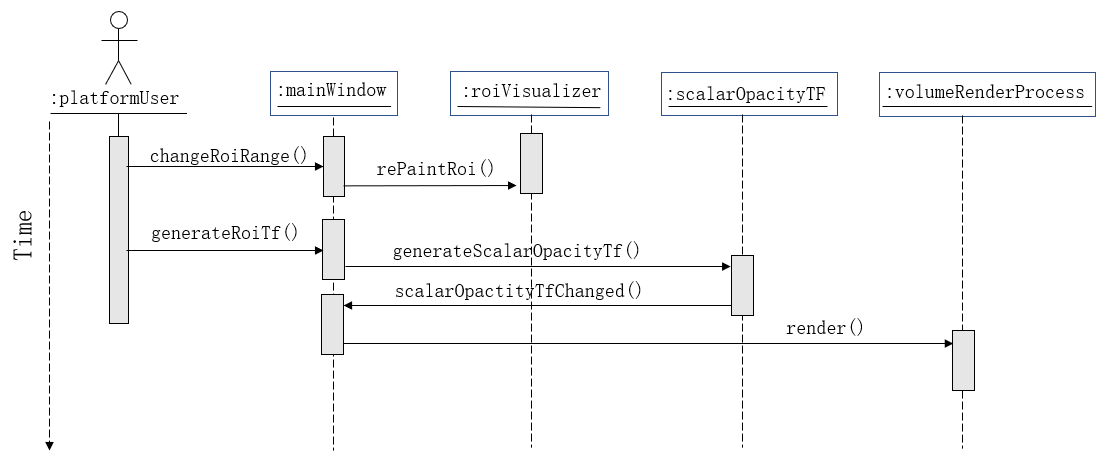


图3.5 根据ROI生成传递函数进行体绘制的顺序图。

第四章 数据读取模块与三维体绘制模块

三维体绘制的对象是三维体数据场，本平台是通过数据读取模块来解析用户传入的DICOM格式图像序列或者NIFTI格式的文件，并构建三维体数据场，然后通过三维体绘制模块来构建三维可视化流程。本章将首先介绍数据读取模块中，所支持的DICOM文件与NIFTI文件的格式标准，然后介绍三维体绘制模块中基于VTK的绘制流程以及增量绘制的优势与实现方法。

## 4.1 数据格式标准

### 4.1.1 DICOM文件解析

DICOM全称是Digital Imaging and Commuication in Medicine，即医学影像成像和通讯标准，于1983年由美国放射学会(ACR)与美国电器制造商协会(NEMA)共同制定的一种医学图像格式标准[22]。经过数十年的发展，该标准已经获得了绝大多数医疗设别生厂商的认可，被广泛应用在医学影像的存储与传输过程中。

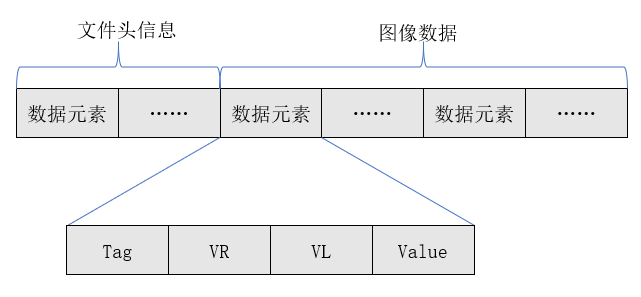


图4.1 DICOM文件格式示意图。

如图4.1所示，DICOM标准的医学图像包含文件头信息以及图像数据两个部分。文件头信息包含Patient，Study，Series等信息；图像数据即是图像的像素值，由实际的测量值转换而来。每一项信息都是以数据元素(data element)的形式存储，每个数据元素又是由以下四个部分组成：

(1) Tag: 标签。由两个十六进制数组成(Group+Element)，标识该数据元素所存储的信息。例如(0010, 0010)表示该数据元素存储的是病人的姓名；

(2) Value representation: 数据类型。表示该数据元素中的数据值的类型以及最大长度。例如“CS”表示字符串，最多16个字符；

(3) Value length: 数据长度。表示该数据元素中的数据值的实际长度；

(4) Value: 数据值。存储该数据元素具体内容。

本平台所能处理的DICOM格式文件基于DICOM3.0标准，文件后缀为.dcm。输入数据时，用户传入一个文件夹，文件夹下包含DICOM图像序列，这些图像来自于同一次测量。平台通过vtkDICOMImageReader读取这些图像序列，并构建起三维体数据场，类型为vtkImageData。例如当读入40张DICOM图像，每一张图像像素都是512\*512时，那么所构建的三维体数据场维度就是512\*512\*40。

### 4.1.2 NIFTI文件解析

NIFTI全称是Neuroimaging Informatics Technology Initiative,于2003年由美国国立卫生研究院(NIH)的数据格式工作组(DFWG)所确定。这种格式是为了解决这之前被广泛使用的ANALYZE 7.5格式文件信息缺失的问题。

NIFIT文件包括文件头信息与文件信息两部分。文件头大小为348字节，这是为了和旧的ANALYZE格式文件保存兼容，头文件信息中包含了例如体素维度、体素偏移量、切片信息等等图像基本信息；文件信息中一共有7个维度，前三维存储空间维度的信息，第四个维度留给时间维度，第五、六、七维度可供用户使用，但第五维度也有一些预定义的用途，如存储特定的体素分布或存储矢量数据。

本平台所能处理的NIFIT格式文件基于NIFIT-1.1标准，文件后缀为.nii。

输入数据时，用户输入一个.nii文件。平台通过vtkNIFTIImageReader读取并解析该文件，构建起三维体数据场，类型为vtkImageData。

## 4.2 三维体绘制

### 4.2.1 绘制流程

本平台的三维体绘制模块负责将数据读取模块传来的vtkImageData类型的三维体数据场进行可视化，通过VTK的可视化管道与绘制引擎来构建可视化的流程，如图4.2所示。该可视化流程有五个重要的部分：

(1) Reader

通过vtkDICOMImageReader或者vtkNIFTIImageReader读入用户输入的DICOM图像序列或者NIFIT格式数据，建立vtkImageData格式的三维体数据；

(2) Mapper

本平台基于vtkGPUVolumeRayCastMapper实现基于GPU的并行光线投射算法，算法的执行会延迟到真正的体绘制开始，即render()函数被执行。

(3) Volume

Volume中的VolumeProperty用来设置立体的各种可视化属性，决定最终可视化的效果，主要包括颜色传递函数、标量-不透明度传递函数以及梯度-不透明度传递函数的设置，由传递函数设计模块控制；也包括阴影、环境光照(ambient)、漫反射光照(diffuse)以及镜面光照(specular)的设置；

(4) Render

Render接收Volume的输出，开始可视化过程，可以调节相机与光线属性；render()函数在这一步执行，开始数据的计算过程。

(5) RenderWindow

RenderWindow定义绘制窗口，显示最终的绘制图像，本平台使用QVTKWidget作为绘制窗口；RenderWindow也控制着用户对绘制图的交互行为，本平台提供的体绘制图交互包括：鼠标右键进行旋转以及鼠标左键进行缩放。

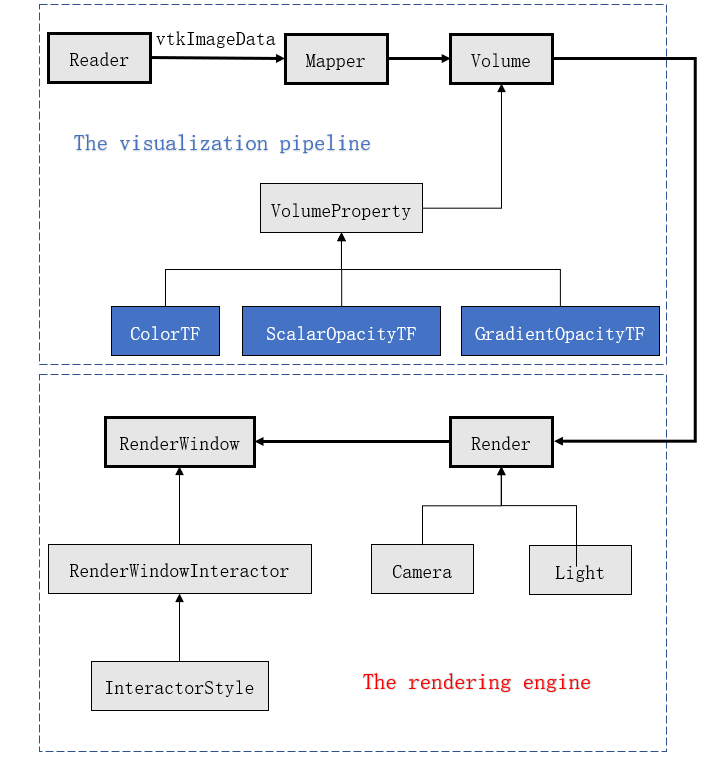


图4.2 基于VTK的三维可视化流程。

通过Reader、Mapper、Volume、Render以及RenderWindow这五个Filter即可实现最基本的三维体数据的可视化。该可视化流程具备VTK的lazy evaluation机制，因此每次体绘制只需要在搭建一次该流程，之后对流程中的某一个步骤进行修改后，只需要重新render即可。例如，对颜色传递函数colorTF进行修改后，只需重新render就可以在最终的绘制图上观察到相应的颜色变化。这种机制为传递函数的实时交互设计提供了可能，即用户对VolumeProperty做的任何更改，都可以实时地在绘制图上观察到相应的变化。

### 4.2.2 增量绘制

本平台的一个重要设计思想是分割与融合。分割即是指在传递函数设计过程中，一次只针对一个感兴趣区域进行设计；融合是指将多个感兴趣区域进行增量绘制，在最终的绘制图上实现叠加。这种设计思路能够有效减低传递函数的设计复杂度。

对于复杂的三维体数据场，若想要将多个感兴趣区域以不同的样式同时绘制出来，只通过一次传递函数设计往往很难实现。例如，对于一个人体头部体数据，需要同时绘制出头骨以及皮肤，那么用户基于标量值选择感兴趣区域时，区域范围会包含骨头与皮肤，也会包含其他标量值属于该范围之间的组织，如肌肉。传递函数设计时，首先需要在用户选定的范围内中定位到骨头与皮肤，然后用不同的颜色以及不透明度表示。

而具备增量绘制的功能后，用户可以先选择骨头作为感兴趣区域，赋予某一颜色与不透明度进行绘制，然后再选择皮肤作为感兴趣区域，赋予另一颜色与不透明度进行绘制，然后将两次的绘制图进行叠加，如图4.3所示。与前一种传统设计方法相比，分割与叠加设计方法有以下三个优势：

(1)用户所选区域即可视区域

即使需要绘制多个感兴趣区域，用户每次也只需要选择其中一个区域，因此传递函数设计时所关注的对象就是用户所选的区域，不需要额外地进行区域分割，减少了交互式传递函数设计的交互复杂度，也减少了数据分析时的计算量。

(2)可实现对每个感兴趣区域的独立控制

由图4.2可知，一个可视化流程中，确定了体数据与传递函数，即确定了体绘制图。每个区域的原始体数据相同；每次传递函数设计只针对多个感兴趣区域中的一个区域，因此可以保证每个区域都有对应的一组传递函数。将这些传递函数保存，就能对多个区域的可视化实现隔离。叠加绘制后可以再对其中某一个区域进行调整，而不影响其他区域的可视化效果。

(3)可实现多个体数据的增量绘制

与多个感兴趣区域的增量绘制类似，多个体数据增量绘制也是每次绘制一个体数据或者一个体数据的一个感兴趣区域，将该此绘制的原始体数据以及传递函数保存，最后实现多个区域的叠加绘制，并对每个区域实现独立控制。多个体数据增量绘制如图4.4所示。

|  |  |  |
| --- | --- | --- |
|  |  |  |
| (a) 头骨绘制图 | (b) 皮肤绘制图 | (c) 头骨与皮肤叠加绘制 |

图4.3 头部体数据骨头与皮肤叠加绘制示意图。

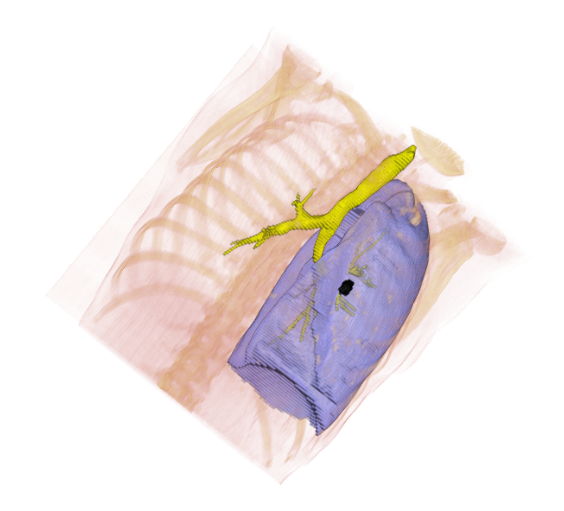


图4.4 四个体数据叠加绘制示意图。淡黄色半透明的是原数据，深黄色的是气管，蓝色半透明的是左肺，黑色的是肿瘤。

增量绘制功能不仅仅只是将多个区域叠加在最终的绘制窗口上，还需要保证叠加之后每个区域的可视化效果与单独绘制时相比没有差异，同时还需要保证多个区域的空间位置在叠加后不会错乱。在增量绘制的具体实现上，VTK中并没有直接提供增量体绘制的功能，但有以下几种方法可以实现叠加绘制：

(1) 使用多个可视化管道

一个Render可接受多个Volume，因此可以将图4.2所示的可视化流程中的可视化管道扩充为多条，每一条管理一个绘制区域的体数据与传递函数，最后通过同一个Render在同一个RenderWindow上绘制。但使用这种方法时，每个体数据都会执行一次光线投射算法，因此在计算对屏幕像素的贡献时，后添加的体数据的贡献会覆盖掉先添加的体数据的贡献，直观表现就是后添加的体数据总是出现在先添加的体数据的前面，存在空间错位，如图4.5(a)所示。

(2) vtkMultiVolume

vtkMultiVolume可将多个体数据进行融合，作为一个体数据去进行绘制。但VTK提供的这种方法目前还存在Bug，即无法对融合后的体数据添加阴影效果，导致叠加绘制与单独绘制的可视化效果存在差异，如图4.6(b)所示。

(3) 多通道vtkImageData

创建一个多通道的vtkImageData，每个区域对应的体数据作为该vtkImageData的一个通道，为Volume添加传递函数时，声明该传递函数作用的通道id。这个方法可以有效解决叠加绘制时的空间错位与可视化效果改变的问题，该平台的增量绘制通过这种方法来实现。

第五章 数据分析模块

增量绘制功能使得传递函数设计或数据分析只需要针对某一个感兴趣区域来进行，该区域是体数据标量值上的一段连续区间。但是当体数据较为复杂或者噪声较多时，该区间可能包含感兴趣区域之外的数据，例如当用户选择标量值区间作为感兴趣区域范围时，若只是单纯地在标量-不透明度传递函数上，将内设为可见，将该区间之外设为不可见，往往无法达到预期的可视化效果，因为可能会有以下两种情况：用户选择的感兴趣区域不精确，实际的感兴趣区域范围是的子区间；其他材质的标量值范围与区间有交集。往往两种情况会同时出现。这时候只通过交互式设计就难以设计出只可视化感兴趣区域的传递函数，需要借助数据分析。本章将介绍边缘检测与K-Means聚类两种数据分析方法。

## 5.1 3D边缘检测

体数据的边缘即是多种材质的交界面,不同材质的标量值往往不同，因此在材质的交界面，标量值的变化会比较剧烈，因此梯度值会较大，3D边缘检测就是根据体数据的3个维度的梯度值来检测边缘。检测结果可以用来设计梯度-不透明度传递函数，与标量-不透明度传递函数结合，来提高可视化效果。

### 5.1.1 检测流程

3D边缘检测的流程与Canny边缘检测[23]类似，本平台的边缘检测是为了指导传递函数生成，因此简化了双阈值分割步骤，也省去了最后一步滞后边界跟踪。本平台的3D边缘检测有以下几个步骤：

1. 高斯滤波

高斯滤波的作用是去除噪声，因为噪声点的梯度值往往也会很大，可能会造成伪边缘。对体数据进行高斯滤波需要立体高斯核，例如半径为1的高斯核就是一个有27个元素的立方体，将该核与原数据的相同大小区域进行三维卷积即可实现高斯滤波。图5.1所示是原体数据切片与经过高斯滤波之后的相同位置的切片。

|  |  |  |  |
| --- | --- | --- | --- |
| (a)原体数据切片 | (b)高斯滤波后的切片 | | (c)梯度幅值可视化 |
| (d)非极大值抑制 | | (e)双阈值分割 | |

图5.1 边缘检测流程。

2. 梯度计算

计算高斯滤波之后的体数据的各点的梯度幅值，本平台采用的是中心差分法来计算梯度，如式(2.8)与式(2.9)所示。图5.1(c)所示是梯度幅值的可视化结果，其中亮度越高说明梯度越大。

3. 非极大值抑制

通过梯度幅值已经能够初步地辨识出图像的边缘，但是这些边缘比较粗，应用非极大值抑制可以将边缘细化。非极大值抑制是将每个边缘点与其邻接的边缘点比较，若该边缘点是局部最大，则保留，否则将该点抑制，即设为0。非极大值抑制能够提取出梯度变化最锐利的位置，可视化效果如图5.1(d)所示。

4. 双阈值分割

通过非极大值抑制能检测出用户设定的标量值范围内的所有边缘，再通过双阈值分割进一步提取处于梯度处于双阈值之间的边界点。图5.1(e)是经过双阈值分割后的可视化边界，与图5.1(d)相比边界有所减少。

### 5.1.2 传递函数生成

## 5.2 K-Means聚类

### 5.2.1 聚类流程

### 5.1.2 传递函数生成

第六章 传递函数设计模块

## 5.2 传递函数设计方法

第六章 总结与展望

在计算机断层扫描(CT)中，仪器测量得到的CT值是由各种组织对X射线的线性吸收系数(μ值)决定的。将测量结果保存为DICOM图像时，图像数据部分的数据值是CT值转换得到的像素值，转换公式如下：

|  |  |  |
| --- | --- | --- |
|  |  | (2.13) |

在VTK中，传递函数是通过分段函数来设置的，用户只需定义几个传递函数的控制点即可，例如，要使用三角函数映射规则时，只需依序传入3个控制点，，即可，控制点之间的函数值通过插值计算得到。

增量绘制能降低传递函数设计的复杂度，并减少数据分析的计算量。

# 参考文献：

[1] 唐泽圣.三维数据场可视化.北京：清华大学出版社，1999，1

[2] Levoy M. Display of Surfaces from Volume Data[J]. IEEE Computer Graphics & Application, 1988, 8(3):29-37.

[3] 范翠萍. 体绘制多维传递函数的设计方法研究[D].江苏科技大学,2011.

[4] Huang, Jian, Mueller, Klaus, Crawfis, Roger, Bartz, Dirk, Meissner, Michael. A Practical Evaluation of Popular Volume Rendering Algorithms[P]. Volume Visualization, 2000. VV 2000. IEEE Symposium on,2000.

[5] 张尤赛,陈福民.三维医学图像的体绘制技术综述[J].计算机工程与应用,2002(08):18-19+122.

[6]. Kniss, J., Kindlmann, G., Hansen, C.. Interactive volume rendering using multi-dimensional transfer functions and direct manipulation widgets[P]. Visualization, 2001. VIS '01. Proceedings,2001.

[7] H. Pfister , B. Lorensen, C. Bajaj. The transfer function bake-off[J]. IEEE Computer Graphics and Applications,, 2001, 21(1):16-22

[8] 周芳芳,樊晓平,杨斌.体绘制中传递函数设计的研究现状与展望[J].中国图象图形学报,2008(06):1034-1047.

[9] J. Marks et al., “Design Galleries: A General Approach to Setting Parameters for Computer Graphics and Animation,” Proc. Siggraph 97, ACM Press, New York, Aug. 1997, pp. 389-400.

[10] Li L, Peng H, Chen X, et al. Visualization of boundaries in volumetric data sets through a what material you pick is what boundary you see approach[J]. Computer methods and programs in biomedicine, 2016, 126: 76-88.

[11] Kindlmann G, Durkin J W. Semi-automatic generation of transfer functions for direct volume rendering[C]//IEEE Symposium on Volume Visualization (Cat. No. 989EX300). IEEE, 1998: 79-86.

[12] Tzeng F Y, Ma K L. A cluster-space visual interface for arbitrary dimensional classification of volume data[C]//Proceedings of the Sixth Joint Eurographics-IEEE TCVG conference on Visualization. Eurographics Association, 2004: 17-24.

[13] Tzeng F Y, Lum E B, Ma K L. A novel interface for higher-dimensional classification of volume data[C]//Proceedings of the 14th IEEE Visualization 2003 (VIS'03). IEEE Computer Society, 2003: 66.

[14] Cheng H C, Cardone A, Jain S, et al. Deep-learning-assisted volume visualization[J]. IEEE transactions on visualization and computer graphics, 2019, 25(2): 1378-1391.

[15] Max N. Optical models for direct volume rendering[J]. IEEE Transactions on Visualization and Computer Graphics, 1995, 1(2): 99-108.

[16] Moller T, Machiraju R, Mueller K, et al. Evaluation and design of filters using a Taylor series expansion[J]. IEEE transactions on Visualization and Computer Graphics, 1997, 3(2): 184-199.

[17] Neumann L, Csébfalvi B, König A, et al. Gradient estimation in volume data using 4D linear regression[C]//Computer Graphics Forum. Oxford, UK and Boston, USA: Blackwell Publishers Ltd, 2000, 19(3): 351-358.

[18] Lum E B, Ma K L. Lighting transfer functions using gradient aligned sampling[C]//IEEE Visualization 2004. IEEE, 2004: 289-296.

[19] Jerrold E. Marsden and Anthony J. Tromba. Vector Calculus, chapter 2.6, 4.2. W.H. Freeman and Company, New York, 1996.

[20] Sato Y, Westin C F, Bhalerao A, et al. Tissue classification based on 3D local intensity structures for volume rendering[J]. IEEE Transactions on visualization and computer graphics, 2000, 6(2): 160-180.

[21] Kniss J, Premoze S, Ikits M, et al. Gaussian transfer functions for multi-field volume visualization[C]//Proceedings of the 14th IEEE Visualization 2003 (VIS'03). IEEE Computer Society, 2003: 65.

[22] 毛淑华，雷伯录. DICOM标准及应用[J]. 科技广场，2006(11):60-62.

[23] J. Canny, A computational approach to edge detection, IEEETrans. Pattern Anal. Mach. Intell. (6) (1986) 679–698.