面向多角色开发的三维 CAD 内核开放架构

梁聪, 徐延宁*, 王璐, 孟祥旭

(山东大学软件学院　济南　250101)
(xyn@sdu.edu.cn)

摘　要: 三维 CAD 内核支撑大型工业软件的开发与应用, 开放的三维 CAD 内核架构便于更多角色参与到 CAD 内核优化、系统开发以及深层次应用中. 围绕三维 CAD 内核开放架构, 文中在剖析主流三维 CAD 内核的基础上, 提出了一种支持多角色协同参与研发的 CAD 内核开放架构, 包括几何表示、拓扑表示、离散化、可视化等模块; 讨论了开放架构的主要功能模块设计、功能模块间互操作以及基于内核的 CAD 应用快速开发等关键问题; 通过基于开放架构扩展开发管线 CAD 桌面应用, 对开放架构的可用性进行了验证.

关键词: 三维 CAD 内核; 开放架构; 建模与渲染
中图法分类号: TP391.72　　**DOI:** 10.3724/SP.J.1089.2023.2023-00002

An Open Framework of 3D CAD Kernel for Multi-Role Development

Liang Cong, Xu Yanning*, Wang Lu, and Meng Xiangxu

(*School of Software, Shandong University, Jinan*　250101)

Abstract: The 3D CAD kernel supports the development and application of large industrial software, and the open framework of 3D CAD kernel supports more roles to participate in CAD kernel development, optimization, and usage. Based on the analysis of mainstream 3D CAD kernels, we propose an open CAD kernel framework that supports multi-role collaborative development, including modules such as geometric representation, topological representation, discretization, and visualization. Key issues of the open framework, such as the object-oriented extension method of CAD main modules, the interoperability between modules and the customized development of applications based on CAD kernel, are discussed; The feasibility of the framework is verified by a customized pipeline-CAD application which is developed by extension of several modules of the framework.

Key words: 3D CAD kernel; open framework; modeling and rendering

数字化设计与制造工作越来越多地依赖于 CAD 应用, 包括 UG 和 Pro/E 等成熟的商用 CAD 应用以及 OnShape 等在线云 CAD 应用. 三维 CAD 内核定义了 CAD 图形数据的表示, 并融合了几何建模、图形绘制等算法, 为 CAD 应用提供了基础支撑. CAD 内核的质量很大程度上决定了 CAD 应用处理数据的规模、效率、精度以及稳定性. 目前商业领域的主流三维 CAD 内核包括达索的 ACIS 和西门子的 Parasolid. SolidWorks, SolidEdge 和 ANSYS 等采用了 Parasolid 内核, AutoCAD, CATIA 和 Pro/E 等采用了 ACIS 内核. 在开源领域较为著名的 CAD 内核是 Matra 的 OCC(Open CASCADE),

收稿日期: 2023-03-24; 修回日期: 2023-09-07. **基金项目**: 国家重点研发计划(2020YFB1708904). **梁聪**(1999—), 男, 硕士, 主要研究方向为三维 CAD 内核架构; **徐延宁**(1975—), 男, 博士, 教授, 硕士生导师, 论文通信作者, 主要研究方向为计算机图形学; **王璐**(1981—), 女, 博士, 教授, 博士生导师, CCF 会员, 主要研究方向为计算机图形学; **孟祥旭**(1962—), 男, 博士, 教授, 博士生导师, CCF 会员, 主要研究方向为计算机图形学.

FreeCAD 和 AnyCAD 等 CAD 应用均是基于 OCC 内核构建的.

基于商用内核的 CAD 应用开发通常面临一些问题: CAD 内核的使用需要商业授权; 不同 CAD 内核的接口差异显著; 难以对 CAD 内核进行定制扩展. 三维 CAD 内核开放架构的意义在于构建开放的环境, 以支持不同角色参与 CAD 内核的开发与应用流程, 并在内核扩展、算法优化以及应用开发等不同层面做出贡献, 共同推动 CAD 开发与应用的进程.

本文在分析了主流 CAD 内核的架构与模块的基础上, 设计了开放架构的层次与模块, 以及层次与模块间的数据和操作接口; 最后, 通过基于开放架构的应用开发实例验证架构的可行性.

1　相关工作

CAD 内核的历史可以追溯到 20 世纪 70 年代. 如图 1 所示, 1973 年, 剑桥大学的 Alan, Charles 和 Ian 联合创办了 Shape Data 公司, 开发并发布了 CAD 实体造型软件 Romulus. 1981 年, Evans Sutherland 收购了 Shape Data 公司, 研发并推出了 Parasolid. 1988 年, UG 收购了 Parasolid 并将其作为内核支撑 CAD 应用的开发.

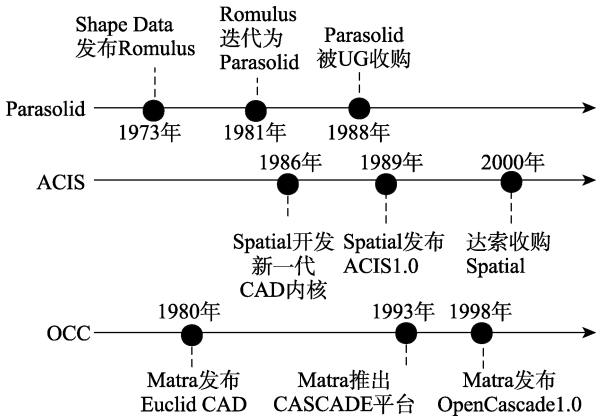


图 1　国外主流 CAD 内核发展历程

ACIS 内核的开创与 Parasolid 有着密切关联. 1986 年, Spatial Technology 公司邀请 Parasolid 的开创者研发新一代面向对象的 CAD 内核. 1989 年, 以 3 位主要贡献者 Alan, Charles 与 Ian 命名的 ACIS 内核发布. 2000 年, 达索公司收购 Spatial Technology, 并推动 ACIS 内核不断完善. ACIS 与 Parasolid 逐步成为工业界影响力最大的三维 CAD 内核.

OCC 是开源领域影响力最大的三维 CAD 内核. Matra 公司于 1980 年发布了 Euclid CAD 系统, 并于 1993 年推出 CASCADE 开发平台, 为 Euclid CAD 提供了基础支撑. 1998 年, Matra 公司基于 CASCADE 平台推出 OCC 内核, 按照开源协议 LGPL(lesser general public license)进行管理, 为 CAD 内核相关的学习、研究以及应用提供服务.

我国也十分注重 CAD 内核及应用的研发. 如图 2 所示, 1987 年, 唐荣锡教授团队设计开发了多面体实体造型系统 PANDA, 之后的高华 CAD、白玉兰 CAD 和 GEMS 等都是较为成熟的 CAD 应用. 进入 21 世纪后, 华天于 2008 年与 UEL 合作开发了 SINOVATION 内核; 中望于 2010 年收购 VX CAD, 并在此基础上开发 Overdrive 内核; 华云于 2021 年基于 DGM 内核推出面向云端协同设计的 CrownCAD.

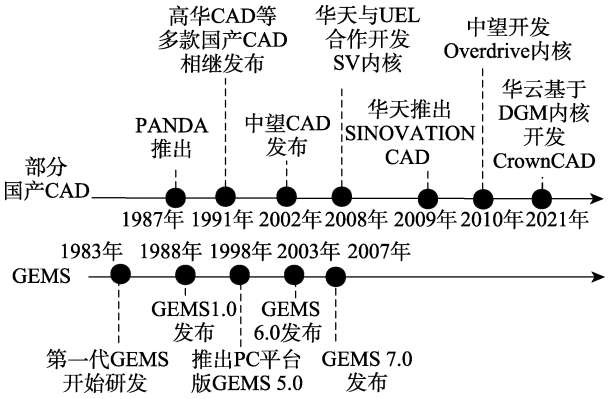


图 2　国内 CAD 应用与内核发展历程

围绕三维 CAD 内核的发展, 华云梅敬成博士提出“CAD 内核这件事情完全不能商业化, 单独研发 CAD 内核是没有太大市场的, 三维 CAD 内核需要与三维 CAD 软件系统整合起来做”. 唐荣锡[1-3]教授提出了“信息公开, 调动一切积极因素”的倡议. 林雪萍[4]提出 CAD 内核是 CAD 应用最核心的基础部分, 对 CAD 应用开发具有重要意义. 何援军[5]分析了国产 CAD 重启的基础, 提出了开放源代码的 CAD 发展策略, 吸引大量开源社群的支持, 更多地发挥广大 CAD 应用开发者的作用.

2　CAD 内核开放架构

CAD 内核为 CAD 应用开发提供支撑, 开放架构支持更多角色共同参与到 CAD 内核的开发与应用过程中. 本节在归纳主流 CAD 内核接口的基础上, 提出一种支持多角色协同参与的开放架构.

2.1　主流 CAD 内核接口

Parasolid 内核通过链接库或组件方式为开发者提供了图 3a 所示的 API 和底层接口 2 类访问方

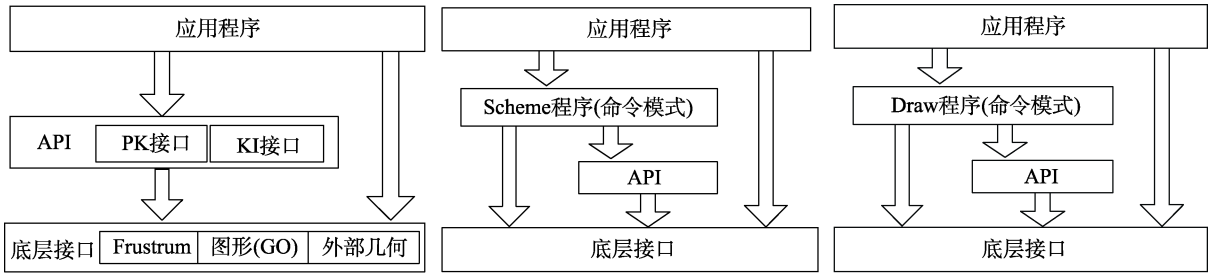


图 3 主流 CAD 内核接口层次对比

式. API 接口 PK(Parasolid kernel)提供几何造型功能, 而 KI(kernel interface)则兼容旧内核版本. 底层接口中的 Frustrum、图形以及外部几何, 提供了包括内存管理、几何计算和文件访问等更深层次的 CAD 数据访问方式. 朱虎等[6]分析了 Parasolid 内核的实体建模、数据读写和可视化等接口, 研究了 Parasolid 的应用开发方法. 郑春峰等[7]基于 Parasolid 内核开发 CAD 系统, 并讨论了构建系统框架、完善建模功能以及建立约束规则的开发步骤, 对 CAD 应用开发过程具有指导意义.

ACIS 不仅支持 API 和底层接口 2 种开发模式, 还提供了图 3b 所示的基于命令行和脚本的开发模式 Scheme AIDE. 开发者通过控制台输入 Scheme 命令, 命令经解释器处理后由 ACIS 内核执行, 通过轻量级可视化界面显示几何造型结果. 万凯等[8]研究了 ACIS 基于底层接口与 API 的开发模式, 构建了 Keren CAD 系统. 吴晓燕[9]面向轴承座的造型设计问题, 基于 Scheme 命令进行快速开发. 张林等[10]通过 ACIS 构建几何模型, 扩展新的渲染引擎, 实现了高质量的零部件渲染.

与 Parasolid 和 ACIS 相比, 图 3c 所示的 OCC 作为开源 CAD 内核, 除了基于底层接口、API 和命令定制开发 CAD 应用外, 还支持对内核中的实体、工具和操作等进行源代码层次的扩展. 赵红红等[11]基于 OCC 的 API 开发了轴三维参数化设计系统, Slyadnev 等[12]基于底层接口开发了检查模型几何和拓扑表示的工具. 刘峥等[13]在源代码层次扩展了建模操作, 支持高效的钣金件建模.

2.2 CAD 内核开放架构设计

图 4 所示的 CAD 内核开放架构作为国家重点研发计划“三维 CAD 几何引擎与研发平台构建”的一部分, 构建在图 5 所示的 CAD 协同研发平台上. 开放架构旨在支撑包括项目团队在内的多角色成员持续参与到内核设计、开发、优化及应用过程中. 基于对主流内核的分析, 以开放性为目标, 设计了架构的层次和模块, 优化改进了数据结构, 支持扩展、定制数据类型, 提高了内核的模型描述能力, 提供了更加完备的功能接口, 包括模型描述、离散化和可视化等, 进一步完善了内核功能域, 并通过构建管线 CAD 应用对架构进行了验证.

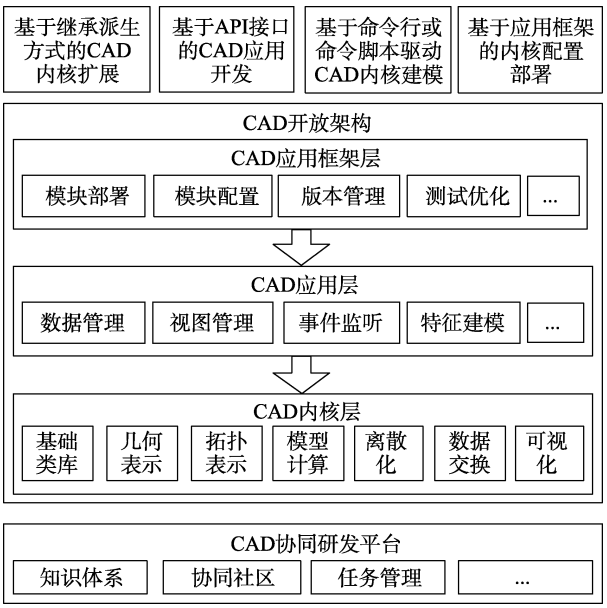


图 4 CAD 内核开放架构

CAD 内核开放架构自下而上分为内核层、应用层和应用框架层: (1) CAD 内核层将 CAD 内核数据及功能模块化组织管理, 包括基础类库、几何表示、拓扑表示、模型计算、离散化、数据交换、可视化 7 个内核模块, 模块间通过接口互操作. (2) CAD 应用层包括使用内核构建 CAD 应用所需的模块, 主要包括数据管理、视图管理、事件监听、特征建模等模块. (3) CAD 应用框架层面向 CAD 应用的敏捷开发与部署, 主要包括模块部署、模块配置、版本管理和测试优化等.

开放架构支持 4 类开发模式: (1) 通过继承派生方式丰富扩展内核中的数据类型与操作; (2) 基于 API 接口调用将 CAD 内核功能集成到应用中;

Application	添加 x86 环境的配置，合并 spline fit 分支的核心代码进入，
DataStructure	添加 x86 环境的配置，合并 spline fit 分支的核心代码进入，
Discrete	集成了b样条曲线相关内容
DiscreteAlgorithm	重构 Mesh3d 结构
Discretization	重构 Mesh3d 结构
Example	添加 x86 环境的配置，合并 spline fit 分支的核心代码进入，
Experiment	添加依赖库 glad 的 release 版本的lib
Foundation	集成了b样条曲线相关内容
GeometricModels	搭建针对曲面拓扑修复的框架，并填充了部分代码
Geometry	集成了b样条曲线相关内容
GeometryAlgorithm	添加 x86 环境的配置，合并 spline fit 分支的核心代码进入，
Math	集成了b样条曲线相关内容
Packages	添加 x86 环境的配置，合并 spline fit 分支的核心代码进入，
Render	添加 x86 环境的配置，合并 spline fit 分支的核心代码进入，
Step	调整 Render 项目代码结构
Test	编写测试类VertexSet3dTest，完成单元测试；修正VertexSet
Topology	同步李亚辉对于 IsSameSense 相关的注释

图 5　CAD 协同研发平台

(3) 命令行或命令脚本驱动 CAD 内核进行参数化建模或批量建模; (4) 基于 CAD 应用框架通过配置、部署等方式敏捷构建 CAD 应用.

3　CAD 内核开放架构主要接口

本节将以几何表示、拓扑表示为例描述内核模块的数据与接口，以离散化与可视化为例讨论内核模块间的互操作，进而讨论 CAD 应用层.

3.1　几何表示

如图 6 所示，几何表示模块中的数据基于面向对象机制描述. 抽象层描述几何表示的公共属性和操作，实现层提供具体的几何表示并实现抽象操作.

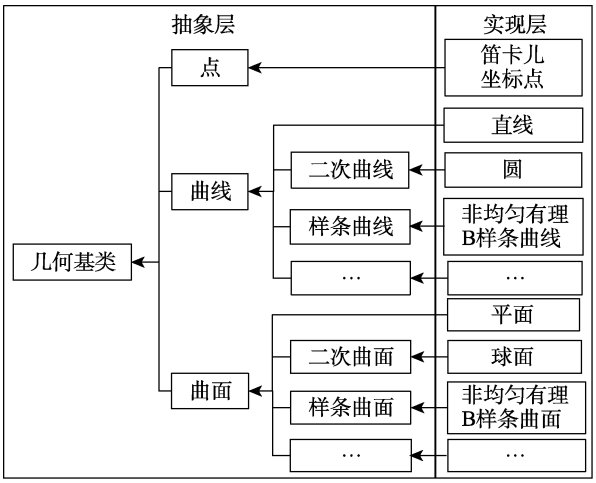


图 6　开放架构几何表示

开放架构中的几何表示为了更好地支撑开发者定制扩展几何元素，应当采用更有层次的继承结构，如图 6 所示. ACIS 和 Parasolid 等采用图 7 所示的扁平的层次结构，预定义丰富的几何类，期望开发者使用预定义几何类并尽可能地避免扩展定制，如 ACIS 直接定义圆、抛物线而省略了二次曲线. 开放架构的几何表示层次结构相对复杂，但提供了更好的扩展性，其引入了更多抽象几何类型以提供扩展功能. 例如，引入二次曲线类型，加入焦点、准线和离心率等描述，开发者可以更方便地扩展定制特殊二次曲线类型，统一管理所有二次曲线类型的几何对象，更深层次地参与到几何类型的扩展中.

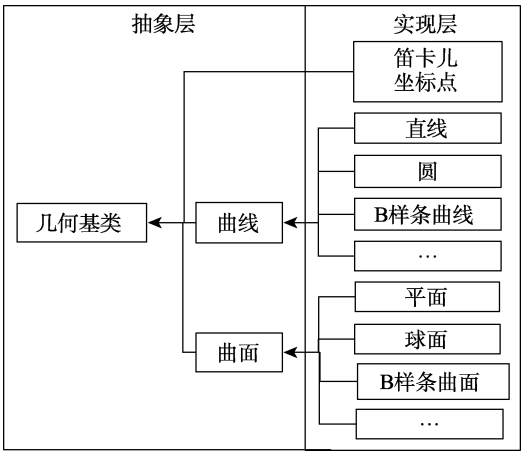


图 7　ACIS 几何表示

为了支撑与其他模块间的互操作，开放架构几何表示的几何基类应明确公共属性和方法. 开放架构在顶层基类中提供了特性描述(如周期性、连续性等)、几何变换、包围体判断和序列化等相关接口，扩展新的几何类型时需要实现这些接口，从而支持与其他模块间的互操作. 例如，拓扑表示模块通过几何表示的包围体接口，快速判断相交情况；数据交换模块调用几何表示的序列化接口，实现流式传输.

几何表示提供了几何表示分析、几何映射和几何拟合操作. 几何表示分析包括判断特性、计算距离等；几何映射包括点到曲线、点到曲面、曲线到曲面的参数映射计算；几何拟合包括曲线和曲面的插值、逼近等.

3.2　拓扑表示

拓扑表示基于边界表示(B-Rep)描述组成模型的几何元素间的层次与连接关系. 如图 8 所示，开放架构的拓扑表示包括基础拓扑元素和扩展拓扑元素.

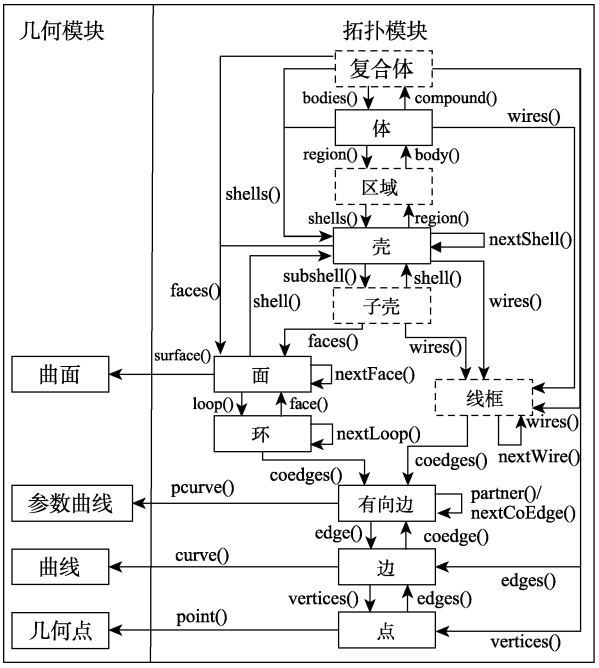


图 8 拓扑结构

结合 Stroud[14]对模型边界表示的描述，拓扑表示应包括表 1 所示的 7 类基础拓扑元素. 为了支持不同类型的拓扑算法，开放架构的拓扑表示允许开发者在基础拓扑元素之外选择添加表 2 所示的扩展拓扑元素，声明模型拓扑结构的层次及拓扑元素类型，定制拓扑表示.

表 1 基础拓扑元素

基础拓扑元素	描述
点	封装了几何点，规定了边的界限
边	几何曲线的有界子集
有向边	含方向信息的边，表示环对边的引用
环	1 组闭合有向边的集合，规定面的界限
面	几何曲面的有界子集
壳	1 组面的集合
体	1 组壳的集合，描述模型整体

表 2 扩展拓扑元素

扩展拓扑元素	描述
线框	1 组不闭合且相连的有向边的集合，支持快速找到 1 组不在环上的边
子壳	壳的分解，用于提高算法的效率，例如，根据子壳的包围盒，加速求交算法
区域	三维空间的连通子集，由 1 组壳规定界限，支持快速判断物体内外关系
复合体	由不同拓扑元素组成的整体，支持描述特殊类型的模型

拓扑表示的操作包括遍历、查找元素以及获取几何表示等. 例如，获取体中某个面的几何信息，可以通过体找到特定的壳，通过壳找到特定的面，最终获取面的几何表示.

3.3 离散化模块

如图 9 所示，离散化模块维护离散模型，支持扩充不同类型的离散数据. 基于离散模型抽象基类派生的子类包括网格数据、多边形数据和离散点数据等，开发者通过派生扩展新的离散表示. 离散化模块支持 3 类操作:

(1) 依据几何和拓扑表示计算离散数据，应用于模型可视化，例如，尤磊等[15]提出一种基于几何特征保持的 CAD 模型网格计算方法，保证了网格模型与几何模型的逼近程度.

(2) 依据离散表示通过简化、细分计算更新其离散数据，例如, Chu 等[16]将 CAD 模型划分为多个细节层次，支持不同细节层次的离散数据间的转换；耿维忠等[17]提出一种基于特征映射的 CAD 模型轻量化方法，提高了模型简化效率.

(3) 依据离散表示通过插值、逼近等方式拟合曲线曲面几何表示，对设备采集的离散数据赋予几何语义，进而支持高效编辑. 例如，柏硌等[18]提出一种基于渐进迭代最小二乘拟合算法的模型实体构建方法，支持将网格模型转化为 T 样条实体.

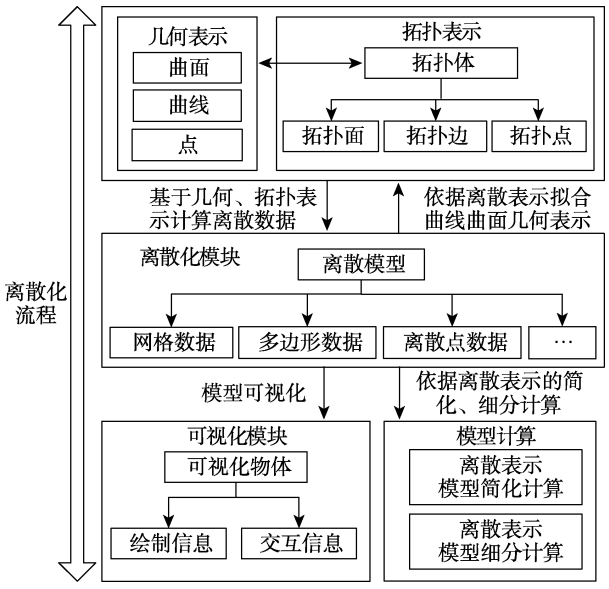


图 9 离散化模块与其他模块间互操作

为了支持离散化算法的应用、迭代和优化，开放架构描述了图 10 所示的通用的离散化流程，开发者可以对流程中的特定步骤扩展优化.

流程的前 2 步包括:

Step1. 创建离散模型. 基于几何表示或离散数据创建相应的离散模型.

Step2. 检查离散模型. 检查离散模型中的几何表示或离散数据是否符合要求.

计算、更新离散表示的操作包括:

Step3. 计算离散表示. 通过边离散化、面离散化算法计算离散数据.

Step4. 检查离散表示. 检查计算的离散表示是否符合要求, 修复结构错误、数据冗余等问题.

基于离散表示的拟合操作包括:

Step5. 计算几何表示. 基于离散模型中的离散数据计算曲线曲面几何表示.

Step6. 检查几何表示. 检查计算的几何表示是否符合要求, 修复自相交等问题.

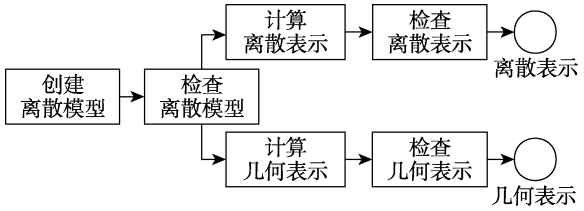


图 10　离散化流程

3.4　可视化模块

如图 11 所示, 可视化模块主要提供模型的绘制、选择功能, 主要包括可视化物体描述、场景数据组织管理以及可视化上下文.

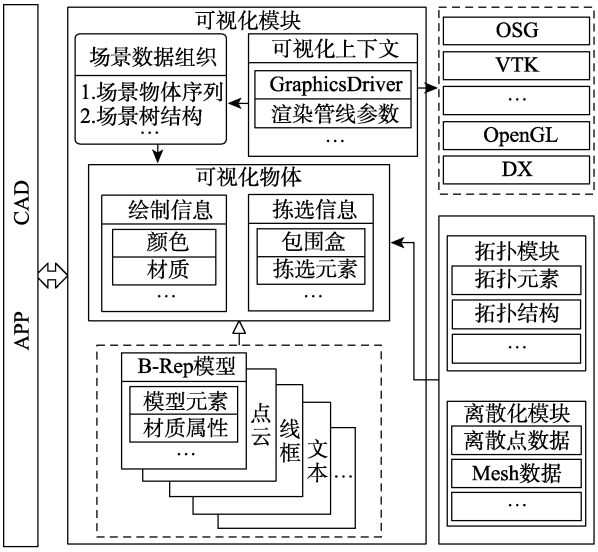


图 11　可视化模块

可视化物体应当具备可以绘制并可以交互选择 2 个重要特征, 因而开放架构描述了 2 类接口: (1) 模型绘制接口. 定义颜色、材质等绘制属性及模型绘制算法, 将离散化模块的数据转化为 OpenGL, DirectX 等绘制引擎可以接受的顶点缓冲对象. (2) 模型选择接口. 定义选择机制及算法, 支持模型及模型拓扑元素的分层选择, 并将元素选中状态通过绘制接口映射为颜色、高亮、透明、视点的组合. 开放架构提供了一组基于可视化接口实现的可视化物体, 包括 B-Rep 模型以及通用点云、线框、文本、坐标轴等, 支持开发者高效开发的同时通过扩展绘制和选择接口定制可视化方式.

场景的组织管理通常基于序列或场景树结构组织可视化物体, 序列结构是一种基础的组织管理方式, 当场景中只有少量物体时便于遍历. 场景树维护了物体之间的层次关系和位置关系, 对树的某一节点施加的空间变换、材质变换等操作, 将作用于子节点, 便于场景管理. 开发者可以扩展场景数据组织算法, 实现可视化物体的自定义管理.

可视化上下文负责可视化的整体流程, 主要处理 3 个方面映射: (1) 连接可视化的场景数据, 通过场景树管理几何的绘制状态; (2) 与 Qt 和 MFC 等窗口和事件系统建立映射, 支持开发者通过配置 3DView 和 TreeView 等构建面向装配、零件编辑以及草图应用的 WorkBench; (3) 与 OpenGL, DirectX 和 OSG 等绘制引擎适配, 控制场景初始化、绘制与刷新. 可视化上下文的主要流程包括:

Step1. 创建 GraphicsDriver, 初始化绘制引擎.

Step2. 创建 Viewer, 注册并绑定场景视图.

Step3. 通过场景数据组织接口访问可视化物体, 获取绘制、交互信息.

Step4. 设置渲染管线参数, 绑定着色器、纹理等.

Step5. 开启绘制引擎消息循环, 监听事件.

3.5　CAD 应用层

如图 12 所示, 应用层提供 CAD 应用程序的开发架构, 整体架构遵循 Model-View-Controller 模式. 其中, View 部分通过视图管理器管理应用视图, 在视图中同步更新绘制结果, 并与 CAD 应用的物理窗口或面板绑定, 支持扩展场景预览、模型编辑、模型装配等典型视图. Controller 部分管理交互事件, 描述相机移动、物体拾取、特征造型等包含 CAD 应用语义的事件. Model 部分组织管理 CAD 数据, 维护 CAD 模型的建模历史数据, 是应用层的核心.

Model 部分主要侧重 3 个方面: 如何描述模型实体, 如何组织和管理实体, 以及如何维护建模历史.

实体在几何、拓扑等信息的基础上, 附加名称、质量等属性以及历史操作数据, 主要有 2 种描述方式. 图 13a 所示的方式 1 将实体设为基类, 包含通用数据和操作, 几何和拓扑作为派生类需要实现数据管理的操作, 模块间耦合过于密切. 图 13b 所示的方式 2 将几何和拓扑等封装为实体类的

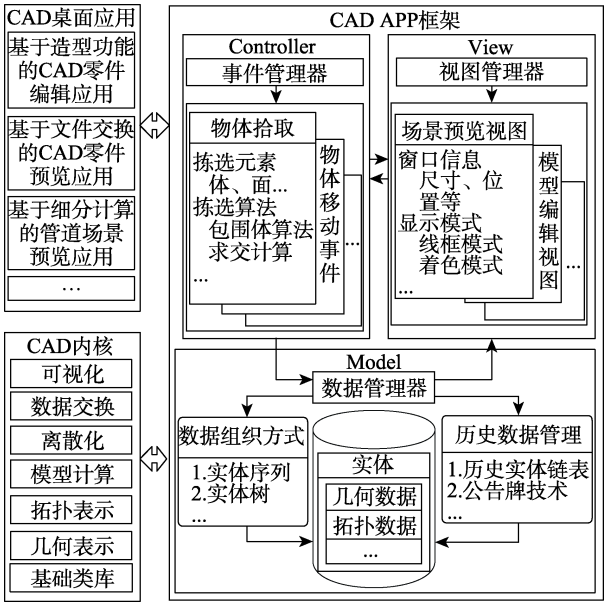
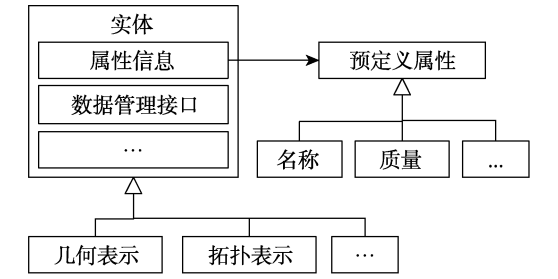
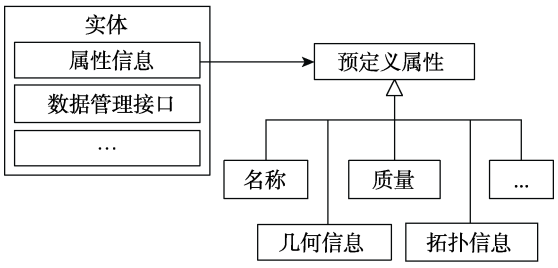


图 12 应用层



a. 实体描述方式1



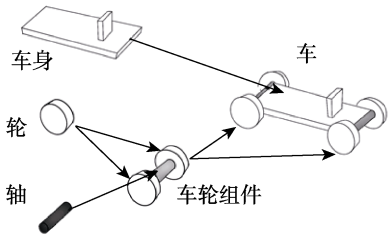
b. 实体描述方式2

图 13 数据管理实体描述方式

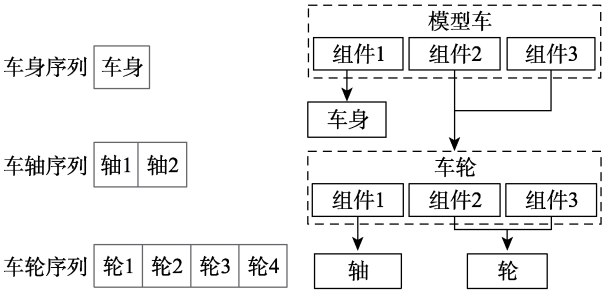
属性，降低了数据管理部分与其他模块间的耦合性. 开放架构中要尽可能地降低模块间的耦合性，因此优先采用第 2 种方式.

实体数据的组织方式包括实体序列和实体树. 前者可以根据实体类型划分多个序列，维护不同类型实体集合. 后者描述了实体之间的层次和装配关系. 图 14 给出了通过实体序列和实体树描述模型车的示例.

Model 中通过图 15 所示的公告板方式维护模型历史记录，支持撤销/重做操作.



a. 模型车及其组件



b. 实体序列 c. 实体树

图 14 模型车的数据组织

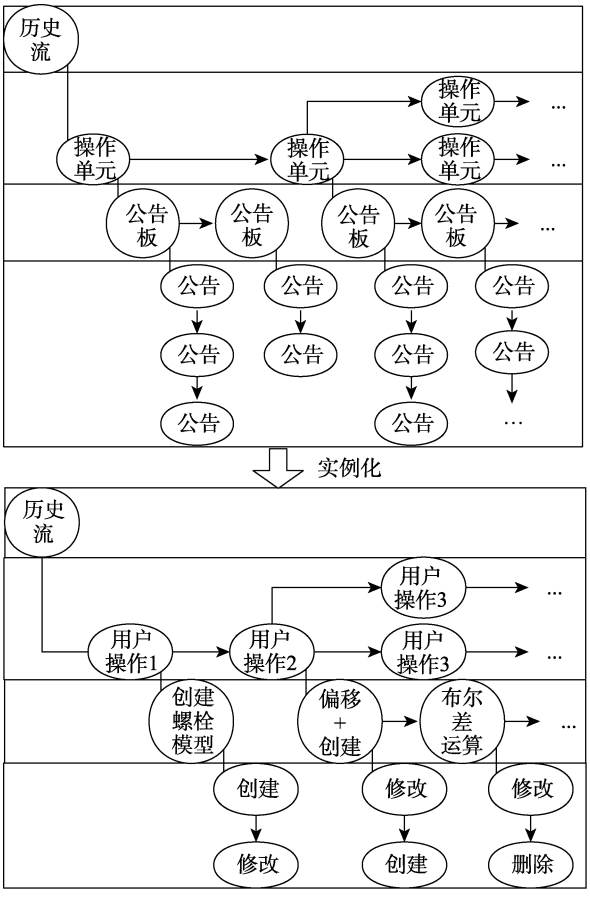


图 15 公告板结构及其实例

公告板包括 4 部分: (1) 公告. 记录实体改变前后的状态，例如，创建、修改和删除等状态. (2) 公告组. 调用建模函数时创建，保存了多个公告，

例如，创建特定模型、执行布尔运算等. (3) 操作单元. 其与用户的每步操作对应，保存了多个公告组，是撤销/重做的基本单位，基于操作单元可以实现不同类型的历史记录，例如，线性历史、分支历史等. (4) 历史流. 用于管理操作单元，保存起始、末尾和当前操作单元. 公告和公告组描述了建模过程的细节，为判断用户意图、模型歧义解决等提供支撑. 在历史数据管理实现过程中，可以进一步扩展，例如，将历史记录与模型共同写入外部文件，从而支持 CAD 多版本建模. CAD 应用执行撤销/重做操作时，在历史流中删除/添加对应的操作单元.

4　基于内核开放架构的应用开发验证

本节基于三维 CAD 内核开放架构的基础数据结构和接口扩展 CAD 内核功能，基于 API 定制开发管线 CAD 桌面应用，以验证开放架构设计. 应用案例基于图 4 所示的应用框架层开发，通过 CMake 构建开发环境，生成模块配置.

4.1　几何表示扩展

在管线 CAD 应用中，开发者通过面向对象的继承方式定制图 6 所示的扩展几何表示，基于细分曲面类派生管线细分曲面，扩充内外半径、端点等公共属性，以及判断内外关系等公共操作. 如图 16 所示，基于管线细分曲面派生直线、贝塞尔曲线管线曲面，设置控制点属性及相关操作，并实现父类抽象操作.

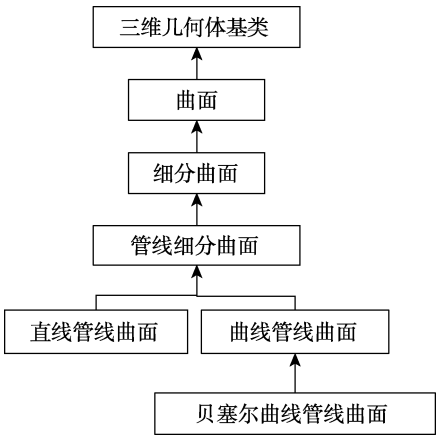


图 16　管线曲面扩展

4.2　拓扑表示扩展

管线 CAD 应用在基础拓扑元素之外添加了区域扩展元素，从而支持管线内外关系的快速判断. 管线模型拓扑表示包括点、边、有向边、环、面、壳、区域和体 8 类元素，其构造方式如图 17 所示.

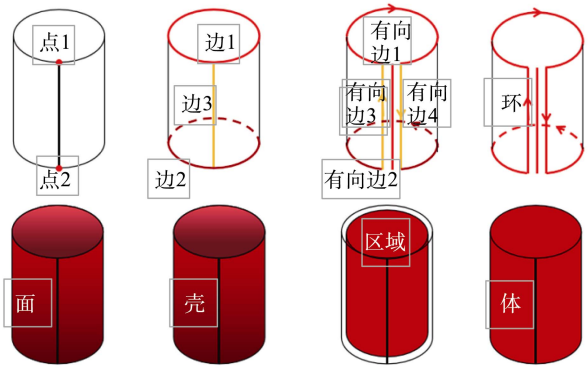


图 17　管线拓扑结构

4.3　离散化模块扩展

管线 CAD 应用基于离散模型类派生细分离散模型，扩充细分次数、精度等属性及相关操作，通过细分算法计算图 18 所示的不同精度的三角形 Mesh 数据. 应用实现了图 10 离散化流程中计算离散表示的各个步骤：基于管线几何表示与拓扑表示中的参数和结构信息，初始化细分离散模型；检查是否存在非法参数等问题，基于半径、控制点等信息进行修复；通过细分算法计算离散数据；检查计算结果中细分面片的数量、精度等是否符合要求.

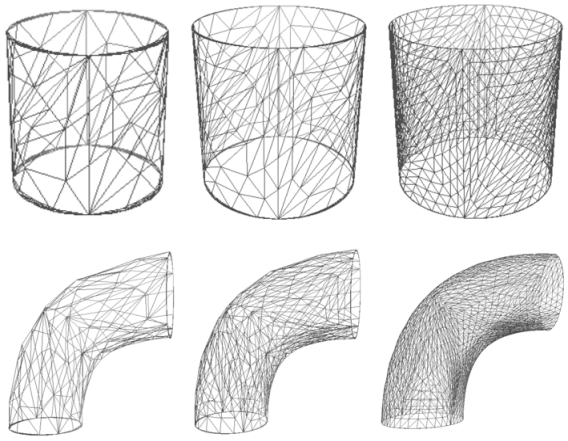


图 18　管线细分计算

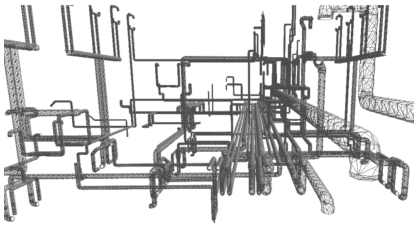
从表 3 可以看出，面向不同规模的 CAD 管线场景，与 OCC 内核的离散化算法相比，管线 CAD 应用的细分离散化算法计算效率更高，且计算得到的网格模型精度更高.

表 3　不同算法管线离散化效率测试

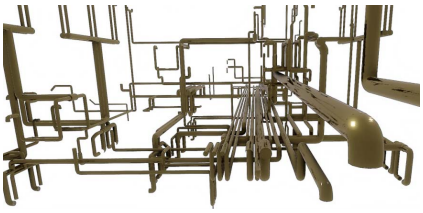
管线数量/个	细分算法		OCC 内核离散化算法	
	时间/s	面片/个	时间/s	面片/个
50	0.0434	54912	0.6075	2096
500	0.4409	800640	6.8362	20488
5000	4.5929	7788342	70.4569	215953

4.4 可视化模块扩展

管线 CAD 应用基于可视化物体基类派生了细分可视化物体和高质量可视化物体，通过细分面片绘制算法和真实感渲染算法实现图 11 所示的绘制接口．可视化上下文根据不同类型可视化物体的参数信息，分别初始化并连接 DirectX 和 DirectEngine 绘制引擎，完成图 19 所示的管线场景绘制．



a. 细分面片绘制



b. 真实感绘制

图 19 管线场景绘制

4.5 CAD 应用模块扩展

CAD 应用模块通过调用内核模块 API，实现应用层功能．数据管理部分通过几何表示、拓扑表示、离散化模块的 API 接口计算几何、拓扑、离散信息，构建管线模型实体，并通过实体树描述了多段管线间的组合关系；视图管理部分通过可视化模块的 API 接口计算管线场景绘制结果，并在场景预览视图中显示；事件控制部分监听用户键盘鼠标操作，通过可视化模块的 API 接口实现相机移动操作．

5 结 语

本文提出一种三维 CAD 内核开放架构，支持多角色参与 CAD 内核的研发应用过程．作为国家重点研发计划项目“三维 CAD 几何引擎与研发平台构建”的一部分，三维 CAD 内核的构建、应用和优化均需要开放架构的支撑．本文讨论了开放架构及其核心模块设计，与主流内核相比，开放架构提供了通过继承扩展数据结构、根据需求定制数据类型、通过不同算法实现功能等开发模式，能够满足内核的开发、扩展需求．基于自主研发的算法和部分算法库，实现了定制开发几何管线 CAD 桌面应用作为框架可行性的验证．桌面应用程序是较传统的应用，如何应用到 Web 服务架构，实现基于开放架构的云 CAD 应用构建，是未来的重点问题．基于开放架构构建云 CAD 应用过程中所涉及的模块化、容器化、服务化等架构问题将是未来讨论的热点．基于开放架构的 CAD 内核多角色协同研发在实践中还需要开发规范和支持版本管理、功能性能测试、模块配置的开发平台的支撑，这些将是后续的研究课题．

参考文献(References):

[1] Tang Rongxi. Review and reflection on the development of CAD industry (I) Dassault/CATIA catch up from behind[J]. Machine Design and Manufacturing Engineering, 2005, 34(1): 40-41(in Chinese)
(唐荣锡. CAD 产业发展的回顾与思考(之一)后来居上的达索/CATIA[J]. 中国制造业信息化, 2005, 34(1): 40-41)

[2] Tang Rongxi. Review and reflection on the development of CAD industry (II) American aircraft company creates CAD brilliance[J]. Machine Design and Manufacturing Engineering, 2005, 34(2): 38(in Chinese)
(唐荣锡. CAD 产业发展的回顾与思考(之二)美国飞机公司创造 CAD 辉煌[J]. 中国制造业信息化, 2005, 34(2): 38)

[3] Tang Rongxi. Review and reflection on the development of CAD industry (III) from romulus parasolid to ACIS[J]. Machine Design and Manufacturing Engineering, 2005, 34(2): 39(in Chinese)
(唐荣锡. CAD 产业发展的回顾与思考(之三)从 Romulus Parasolid 到 ACIS[J]. 中国制造业信息化, 2005, 34(2): 39)

[4] Lin Xueping. When engineers meet mathematicians, open the mystery of CAD kernel[J]. China Equipment: Machine Discovery, 2018(9): 88-92(in Chinese)
(林雪萍. 当工程师遇到数学家打开 CAD 内核的奥秘[J]. 中国机械: 装备制造, 2018(9): 88-92)

[5] He Yuanjun. Road to restart domestic CAD software[J]. Computer Integrated Manufacturing Systems, 2021, 27(11): 3057-3075(in Chinese)
(何援军. 国产 CAD 软件重启之路[J]. 计算机集成制造系统, 2021, 27(11): 3057-3075)

[6] Zhu Hu, Fu Jianhui. A study on the platform for Parasolid application[J]. Machinery Design & Manufacture, 2010(4): 69-70(in Chinese)
(朱虎, 扶建辉. Parasolid 应用平台研究[J]. 机械设计与制造, 2010(4): 69-70)

[7] Zheng Chunfeng, Wang Yongjiao, He Lili. PARASOLID: based CAD system architecture[J]. Application Research of Computers, 2006, 23(8): 216-217, 250(in Chinese)
(郑春峰, 王永皎, 何利力. 基于 PARASOLID 的 CAD 系统开发[J]. 计算机应用研究, 2006, 23(8): 216-217, 250)

[8] Wan Kai, He Yuanjun. Design & development of an advanced 3D CAD system based-ACIS[J]. Computer Simulation, 2004, 21(10): 178-181(in Chinese)

(万凯, 何援军. 基于 ACIS 平台的大型三维 CAD 系统的设计与开发[J]. 计算机仿真, 2004, 21(10): 178-181)

[9] Wu Xiaoyan. Modeling and design of bearing pedestal based on ACIS platform[J]. Mechanical & Electrical Technology, 2011, 34(5): 68-70(in Chinese)
(吴晓燕. 基于 ACIS 平台的轴承座建模设计[J]. 机电技术, 2011, 34(5): 68-70)

[10] Zhang Lin, He Yuanjun, Cai Hongming. Design and development of advanced rendering engine based on ACIS[EB/OL]. [2023-03-24]. https://www.zhangqiaokeyan.com/academic-conference-cn_meeting-28274_thesis/020221981005.html(in Chinese)
(张林, 何援军, 蔡鸿明. 基于 ACIS 的高级渲染引擎的设计与开发[EB/OL]. [2023-03-24]. https://www.zhangqiaokeyan.com/academic-conference-cn_meeting-28274_thesis/020221981005.html)

[11] Zhao Honghong, Wang Hongshen, Zhang Xiangyu. Design of parametric system of shaft based on Open CASCADE modeling[J]. Machine Building & Automation, 2019, 48(4): 97-99(in Chinese)
(赵红红, 王洪申, 张翔宇. 基于 OpenCASCADE 的轴三维参数化设计系统[J]. 机械制造与自动化, 2019, 48(4): 97-99)

[12] Slyadnev S, Malyshev A, Turlapov V. CAD model inspection utility and prototyping framework based on OpenCascade[EB/OL]. [2023-03-24]. https://www.researchgate.net/ publication/319078392_CAD_model_inspection_utility_and_prototyping_framework_based_on_OpenCascade

[13] Liu Zheng, Sun Bo. Research on 3D sheet metal design system based on OpenCASCADE modeling kermel[J]. Machinery Design & Manufacture, 2012(8): 92-94(in Chinese)
(刘峥, 孙波. 基于 OpenCASCADE 内核的三维钣金系统研究[J]. 机械设计与制造, 2012(8): 92-94)

[14] Stroud I. Boundary representation modelling techniques[M]. London: Springer, 2006.

[15] You Lei, Shen Zeyu, Zhang Liqiang, *et al*. A mesh optimization algorithm for CAD models[J]. Journal of Computer-Aided Design & Computer Graphics, 2019, 31(6): 878-885(in Chinese)
(尤磊, 申则宇, 张立强, 等. CAD 模型三角网格优化算法[J]. 计算机辅助设计与图形学学报, 2019, 31(6): 878-885)

[16] Chu C H, Chan Y H, Wu P H. 3D streaming based on multi-LOD models for networked collaborative design[J]. Computers in Industry, 2008, 59(9): 863-872

[17] Geng Weizhong, Yu Jiangrao, Jiao Yuejun. Highly-efficient CAD model simplification method[J]. Mechanical Engineering & Automation, 2022(1): 197-199, 202(in Chinese)
(耿维忠, 于江饶, 焦跃军. 一种高效的 CAD 模型简化方法[J]. 机械工程与自动化, 2022(1): 197-199, 202)

[18] Bo Luo, Zhao Gang, Wang Wei, *et al*. Reconstruction of T-Spline solid from boundary mesh[J]. Journal of Computer-Aided Design & Computer Graphics, 2018, 30(10): 1817-1826(in Chinese)
(柏硌, 赵罡, 王伟, 等. 基于边界网格模型的T样条实体重建[J]. 计算机辅助设计与图形学学报, 2018, 30(10): 1817-1826)