# 第1部分 VTK介绍

## 第1章 欢迎

## 第2章 安装

## 第3章 系统概览

3.1系统架构

Visualization Toolkit由2个基本子系统组成：(1)编译好的C++类库与(2)“解释的”wrapper层—你可以使用Java, Tcl和Python操纵编译好的类。



VTK是面向对象的系统。如果你擅长C++，应用程序完成可以用C++来开发。可以组合对象来构建应用程序。

下面介绍VTK的2个主要部分：可视化管线和渲染引擎。

Low-level Object Model

VTK对象可认为是置于超级类vtkObject中的对象模型。创建和销毁使用New()和Delete()方法。

Reference Counting

Run-time Type information

Object State Display

The Rendering Engine: vtkProp, 。。。

一个简单的例子./VTK/Examples/Rendering/Cxx/Cylinder.cxx

可视化管线

VTK采用数据流(data flow)方法将信息转换为图形数据，该方法涉及2个基本类型的对象：

**•** vtkDataObject

**•** vtkAlgorithm

数据对象表示各种类型的数据，类vtkDataObject可视为通用的数据集。

图3-2显示了VTK支持的数据集对象：

图3-3显示了VTK支持的属性数据。

图3-4显示了可视化管线。



图3-5介绍了一些重要的可视化概念。源算法读取并产生数据或创建一个或对个数据对象。Filter, Mapper, writer



管线执行

VTK可视化管线仅当需要计算的数据时才执行(Lazy evalution)。例子中，实例化一个reader对象，询问很多点。

通常用户不必手动激活Update()，因为过滤器连接到了可视化管线。

图3-6是高级显示管线执行。



管线执行过程详细参考第15章的"管理管线执行"。

3.2开发应用程序

**C++语言**

以Cone.cxx为例，位于：Examples/Tutorial/Step1/Cxx

第1步：使用CMake编译，CmakeList.txt与Cone.cxx在一块，使用FindVTK和UseVTK CMake模块。这些模块会定位VTK，然后设置include的路径，连接并Build C++程序。如果没有成功找到VTK，则必须手动指定合适的CMake参数。



**Python语言**

vtkpython可执行程序

Examples/Tutorial/Step1/Python/Cone.py

执行：vtkpython Cone.py

你可以创建自己的Python脚本程序。

VTK/Examples/Tutorial/Step2/Python/Cone2.py

3.3不同语言之间的转换

不同语言的VTK应用程序之间的转换是直接的，类的名称和方法名称不变。需要修改的是实施细节和GUI界面。例如，C++声明：

anActor->GetProperty()->SetColor(red,green,blue);

用Tcl变成：

[anActor GetProperty] SetColor $red $green $blue

用Python变成：

anActor.GetProperty().SetColor(red, green, blue)

# 第2部分 用例子学习VTK

## 第4章 基础知识

本章的示例代码都是Tcl的，需要研究C++的示例代码。

## 4.1创建简单的模型

4.2使用VTK Interactors

4.3过滤数据

4.4控制相机Camera

4.5控制灯光Lights

4.6控制3D属性

4.7使用纹理

4.8Packing

4.9vtkCoordinates和坐标系统

4.10控制vtkActor2D

4.11文本注解

4.12特殊的绘图类

**Scalar Bar** vtkScalarBar

**X-Y Plots** vtkXYPlotActor

**Bounding Box Axes (vtkCubeAxesActor2D)**

**Labeling Data**

4.13转换数据

## 第5章 可视化技术

5.1可视化vtkDataSet（和子类）

5.2可视化多边形数据

5.3可视化结构网格

vtkStructuredGrid数据集，由六面体(vtkHexahedron)和四边形(vtkQuad)单元组成。

手动创建vtkStructuredGrid

VTK/Examples/DataManipulation/Cxx/SGrid.cxx

提取计算平面

**Subsampling Structured Grids**

5.4可视化矩形网格

手动创建vtkRectilinearGrid（常用于数值分析），由体素(vtkVoxel)或像素(vtkPixel)单元组成。VTK/Examples/DataManipulation/Cxx/RGrid.cxx.

提取计算平面

5.5可视化非结构网格

所有单元类型都能表示为非结构网格。

手动创建vtkUnstructuredGrid

提取部分网格

在非结构网格上做等值线

## 第6章 图像(Image)处理和可视化

## 第7章 体渲染

## 第8章 信息可视化

## 第9章 地球空间可视化

## 第10章 创建模型

## 第11章 时间变化的数据

## 第12章 读写数据

## 第13章 交互、控件和选择

# 第3部分 VTK开发人员手册

# 第14章 贡献代码

# 第15章 管理管线执行

# 第16章 与VTK数据对象的接口

# 第17章 如何编写VTK算法

# 第18章 与窗口系统整合

# 第19章 编码资源