

# ParaView 1.4 使用指南

超级计算中心

中科院计算机网络信息中心

---

1 介绍 .....	3
2 开始 .....	4
2.1 安装 .....	4
Unix 下的安装: .....	4
2 开始 PARAVIEW .....	5
On Unix: .....	5
On Windows.....	5
2.2 术语和概念 .....	6
3 总览 .....	7
3.1 菜单栏 .....	8
3.2 工具栏 .....	9
3.3 左部面板 .....	9
3.4 显示区 .....	10
3.5 状态栏 .....	10
4 实例 1—一个简单的例子 .....	10
4.1 开始 ParaView.....	10
4.2 创建一个球体 .....	11
4.3 改变成为丝形帧 .....	13
4.4 改变决议.....	13
4.5 与球体交互 .....	14
4.6 增加一个包围盒 .....	15
4.7 缩小球体 .....	16

---

4. 8 使用法线来填色 .....	17
<b>5 实例 II—流线可视化 .....</b>	<b>18</b>
5. 1 加载数据文件 .....	18
5. 2 提取子网格 .....	21
5. 3 创建流线 .....	23
5. 4 增加流管 .....	24
5. 5 保存流管 .....	26
<b>6 实例III—动画等高面 .....</b>	<b>26</b>
6. 1 加载数据文件 .....	27
6. 2 创建一个等值面 .....	29
6. 3 剪切等值面 .....	31
6. 4 动画等值面 .....	32
6. 5 删除等高和剪切过滤器 .....	33
6. 6 动画一个剪切面 .....	34
附录 A—文件格式 .....	37
附录 B—数据类型.....	40
附录 C—三维部件 .....	42
附录 D—相机控件.....	43

---

# 1 介绍

ParaView 是对二维和三维数据进行可视化的一种 turnkey 应用。它既可以运行于单处理器的工作站，又可以运行于分布式存储器的大型计算机。这样，ParaView 既可以运行单处理应用程序，又可以通过把数据分布于多个处理器而处理大型数据。ParaView 工程的目的如下所示：

- 开发出一个资源开放、多平台的可视化应用程序；
- 支持分布式计算模型以处理大型数据；
- 创造一个开放的、可行的，并且是直觉的用户接口；
- 开发一个基于开放标准的可扩展的结构

ParaView 使用 VTK 作为数据处理和绘制引擎，并且有一个由 Tcl/Tk 和 C++混合写成的用户接口。这种结构使得 ParaView 成为一种功能非常强并且可行的可视化工具。既然所有的 VTK 数据源和数据处理过滤器要么是立即可以进行访问的，要么是可以写一些简单的构造文来添加的，那么 ParaView 用户就可以使用成百上千的数据处理和可视化算法。另外，使用 Tcl 脚本语言作为核心元素也允许用户和开发人员更改 ParaView 的处理引擎和用户接口来适用本人的需要。

若想更加深入的了解 ParaView，可以访问网站 <http://www.paraview.org> 或是参考ParaView用户使用实例。

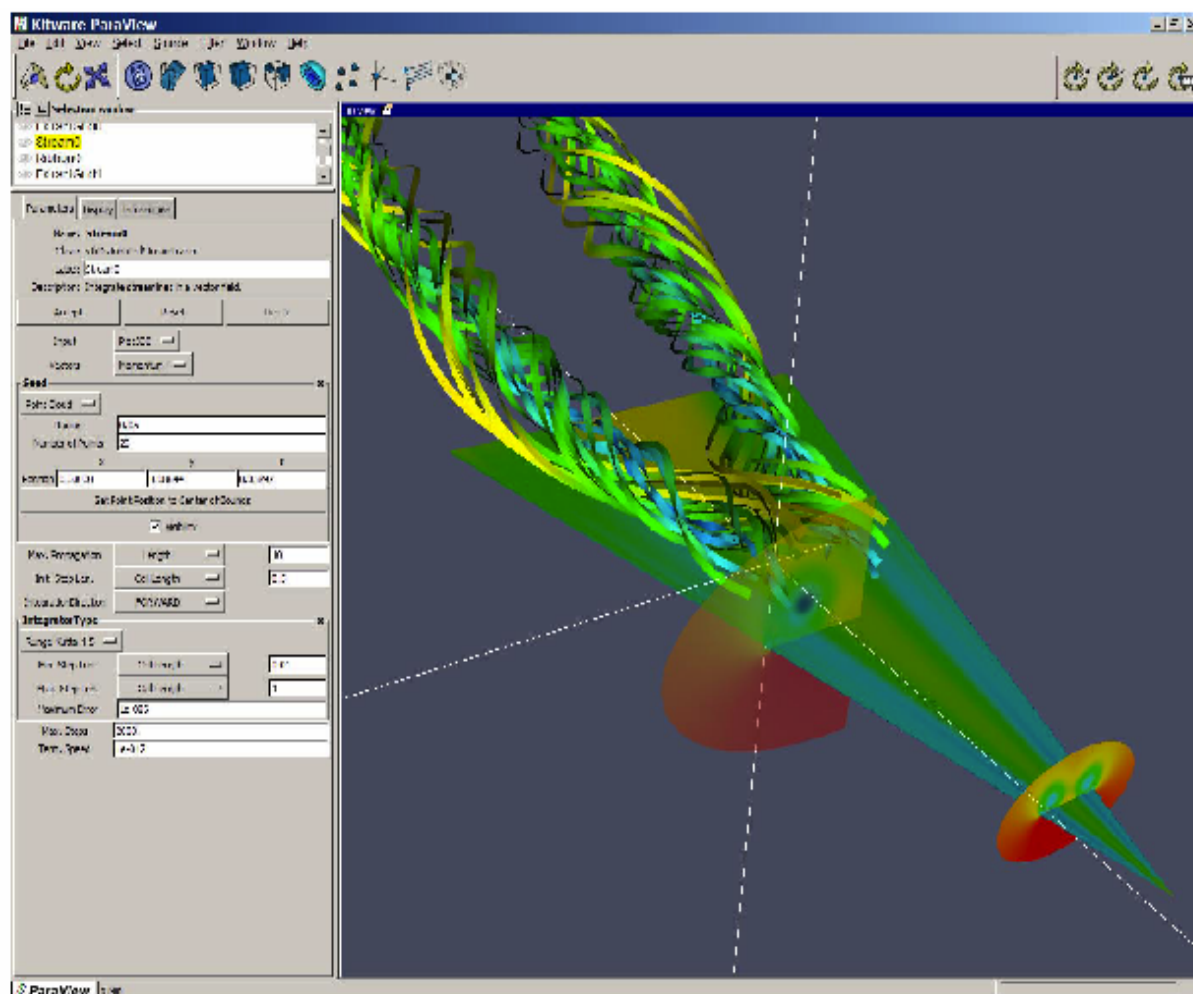


Figure 1. ParaView Screenshot.

## 2 开始

### 2、1 安装

#### Unix 下的安装:

解压缩相关的发布, 如果需要的话, 改变当前路径到你所想的地方。这样就会把二进制数据安装到 `bin/` 目录下, 把文档安装到 `doc/paraview-1.4` 目录下, 并且支持把文档存入 `share/paraview-1.4` 目录。我们推荐安装到 `/usr/local` 或者是 `/opt` 目录下。

#### Windows 下的安装:

---

运行 PVSteup.exe 程序，按照说明进行安装。在默认情况下，将会对 ParaView 进行可执行安装，并且支持在 C:\Program Files\ParaView 1.4 目录下的文档。

## 2、2 开始 PARAVIEW

执行 ParaView 由多种不同的方法，这主要依赖于是否运行于单处理器模式，还是分布式处理器或是客户端/服务器模式。其中，ParaView 最简单的执行模式就是单处理器模式：

### On Unix:

如果 ParaView 被安装于你当前所搜索的路径（\$PATH 或者 \$path），你可以通过指令行来激发 ParaView：

`$paraview`

如果失败的话，你就不得不要么更改搜索路径，要么通过指定可执行的完整路径来运行它。例如，如果安装路径是：/usr/local/bin,那么如下指令将会启动应用程序：

`$/usr/local/bin/paraview`

ParaView 能够识别很多指令行参数。可以通过—help 选项激发 ParaView 而得到可用参数系列。若想得到全部指令行参数和它们的描述，可以参考 ParaView 用户使用实例。

### On Windows

在开始菜单中选择“程序，ParaView 1.0，ParaView”，开始运行 ParaView。

对于怎样在分布式或客户端/服务器模式下运行 ParaView，参考

---

ParaView 用户使用手册。

在运行程序后，可以操作演示程序对该应用程序的应用范围进行可视化介绍。开始演示，选择 Help 菜单下面的“Play Demo”项。

## 2、3 术语和概念

因为 ParaView 在引擎罩下使用 VTK 作为数据处理引擎，文档中包含了一些从 VTK 中引进的概念。其中非常重要的包括：

对象：模拟系统中实体状态和行为的抽象

数据对象：数据抽象。例如，当加载一个 Plot3D 文件后，该文件就通过一个数据对象来代表。

过程对象：一个可视化的对象，也即一个过程或算法的抽象。例如，等值面提取算法就是由一个过程对象来实现。

资源：一个过程对象至少产生一个输出（数据对象）。

reader：从一个或者多个数据文件中读取数据的资源对象。

过滤器：一个至少需求一个输入并且产生一个输出的过程对象。过滤器包括：等高线过滤器，平面截取过滤器，流线过滤器，以及小程度的抽取过滤器等。

模块：在 ParaView 中，过程对象和相关的用于处理这些对象的用户接口成为模块。ParaView 模块包括 EnSightreader 模块（reader），球模块（资源），流线跟踪模块（过滤器）和 ParaView 数据写入模块（writer）。

可视化管道：在 ParaView 中（和在 VTK 中），过程对象互相联系

形成一个可视化管道。管道中的每一个过程对象代表一个对某个数据的操作。

欲求这些概念详细的解释，可以参考 VTK，一个对三位图形进行面向对象处理的方法。

### 3 总览

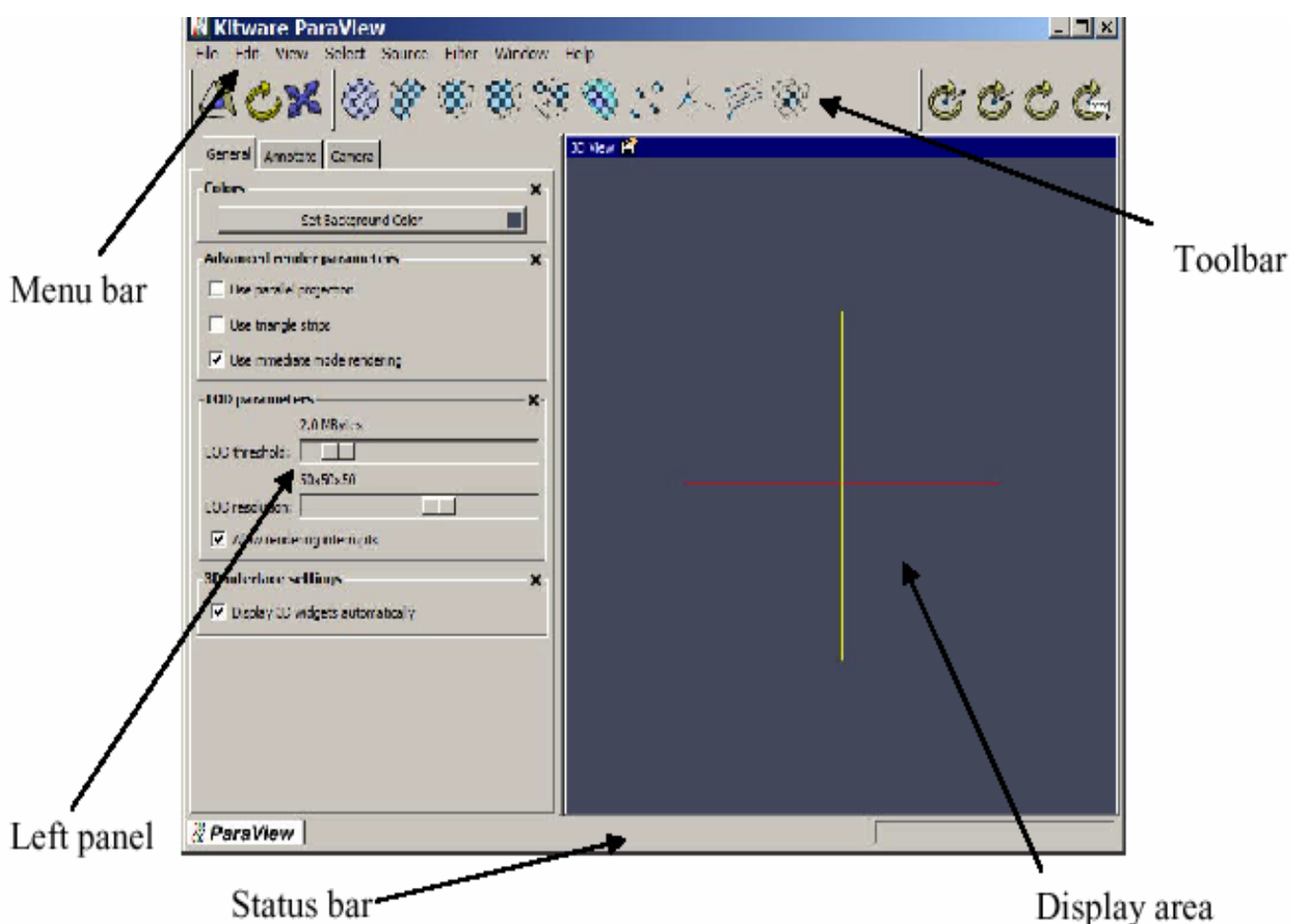


Figure 2. ParaView Window.

在程序开始，ParaView 将会如上所示。有几个区域用作用户接口：在应用程序的顶部是菜单栏，在菜单栏下方是工具栏，在面板左边包括了属性单和挑选/导航窗口，右边则是显示区域，在底层边缘是状态



---

栏。下面对于这些区域将进行更为详细的描述。

### 3. 1 菜单栏

顶部的菜单栏提供了菜单按钮，用于加载和保存数据，改变属性条，产生资源和过滤器，观看其它窗口，显示帮助，和其它一些标准功用。下面对每一菜单按钮进行更为详细的描述。

文件：文件菜单能够被用于加载和保存数据文件，保存批脚本，加载和保存会议文件，保存和打印图像，打开以前所访问过的数据文件和退出应用系统。

编辑：编辑菜单能用于从应用程序中删除所有的数据，和把当前的图像保存到剪贴板（仅在 Windows 操作系统中可用的拷贝）。

查看：查看菜单能够用来选择在应用程序面板左边的属性单。ParaView 所提供的大部分功能都包含在这些属性单中。这里可以设置资源和过滤器的属性，也可以设置应用程序和三位显示比如背景颜色、详细参数的级别以及相机控件的放置和动画的创造。

选择：选择菜单也控制着应用程序左部面板所显示的属性条。选择菜单包含了系统中所加载或创造的每一数据对象的入口。选择其中一项将会引起用户接口左部面板相应的属性条。另外这里也有三个标准的字型资源可以通过字型子菜单来访问。使用选择菜单来选择一个资源就像似于使用选择/导航窗口来选择。

资源：资源菜单包含了一系列可以创建的资源。在选择了一个资源后，改资源的属性条将会显示在属性条区域。设置参数然后按下确认

---

按钮创建资源，或者按下删除按钮来取消操作。

**过滤器：**过滤器菜单包含了一系列的适用于当前所选数据对象的过滤器。通常是最后所创建的数据对象，或者是最近经常从选择菜单或选择/导航窗口所选择的对象。该菜单下可用的过滤器对映于所选择数据类型，并随其改变而改变。一旦选定一个过滤器，该过滤器的属性就将显示出来。设置参数并按下确认按钮即可把它应用于当前所选择的数据对象，或者按下删除按钮而取消该操作。

**窗口：**窗口按钮可以用于显示或隐藏左边面板。另外窗口菜单也可用于产生快捷键、错误日志或定时器日志窗口。

**帮助：**帮助菜单可以用于显示关于 **ParaView** 的帮助和版本信息，也可以进行演示。

## **3. 2 工具栏**

工具栏就位于菜单栏的正下方。工具栏包含了重置相机，切换二维和三维交互模式，和改变旋转中心的按钮。另外，工具栏也包含了一些常用过滤器的按钮。

## **3. 3 左部面板**

左部面板在底部显示属性条，在上部显示选择/导航窗口。而选择/导航窗口只有在数据已经加载或创建后才显示出来。使用菜单栏中的窗口菜单可以隐藏或显示左部面板。左部面板当前的内容是由菜单栏中的查看和选择菜单来控制的，或者通过从选择/导航窗口选择一个数据对象来控制。

---

### 3. 4 显示区

显示区就是场景的三维显示进行绘制的地方。在这个区域，鼠标和键盘可以进行交互。

### 3. 5 状态栏

在应用程序的底部的状态栏将会提供小型的帮助信息，进程更新，和错误提示（在右角落的一个小红声明）。单击错误提示将会产生错误日志，并且把声明的颜色从红改为黑。

## 4 实例 1—一个简单的例子

本节将会展示一个关于 ParaView 的例子，以此使你能够对该应用程序的基本操作熟悉一下。接下来的实例包括了更为详细的例子，如果你对这里展示的基础东西满意的话就可以进行一下尝试。

### 4. 1 开始 ParaView

如果你还未安装 ParaView，那么就请按照向导开始处介绍的安装提示先安装该系统。

在本实例中，我们将会在单处理机模式下运行 ParaView。要开始这种模式下的应用程序，就按照“开始 ParaView”这节的知道开始吧。

本实例假定在你按照下面的指导性步骤之前还未执行任何动作。不然，所创建的模块名字或许会有所变化。

## 4. 2 创建一个球体

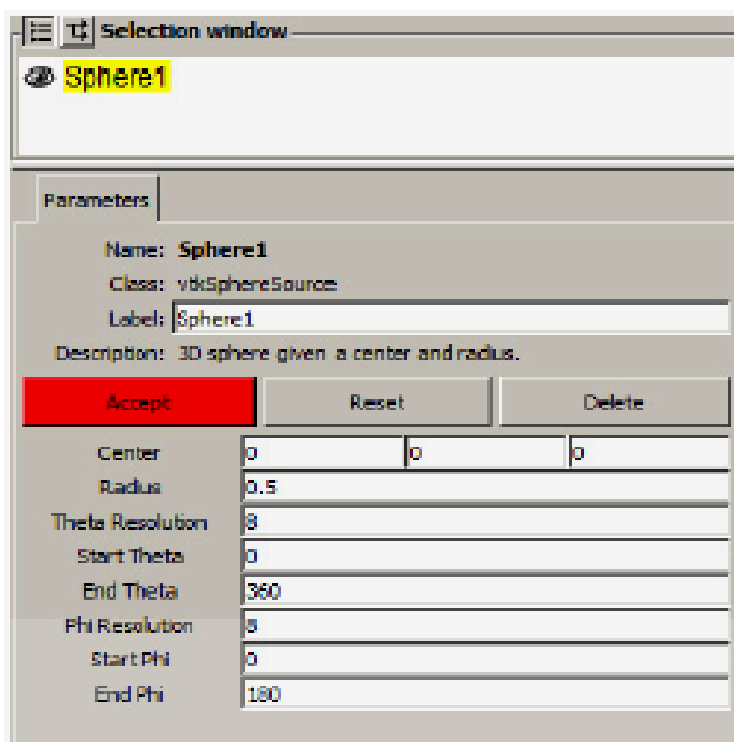
ParaView 开始时是一个空的场景。此时你可以加载数据也可以创建一个资源。资源是用户可配置的对象，可以产生各种类型的数据。这其中包括三维的文本，球体，Mandelbrot 系列。在本例子中，我们将会像右边所示那样，使用菜单栏中的资源菜单创建一个球体。



一旦你选择了球选项，左

边的面板将会改变到在上部显示选择窗口，下部显示球资源的属性条。

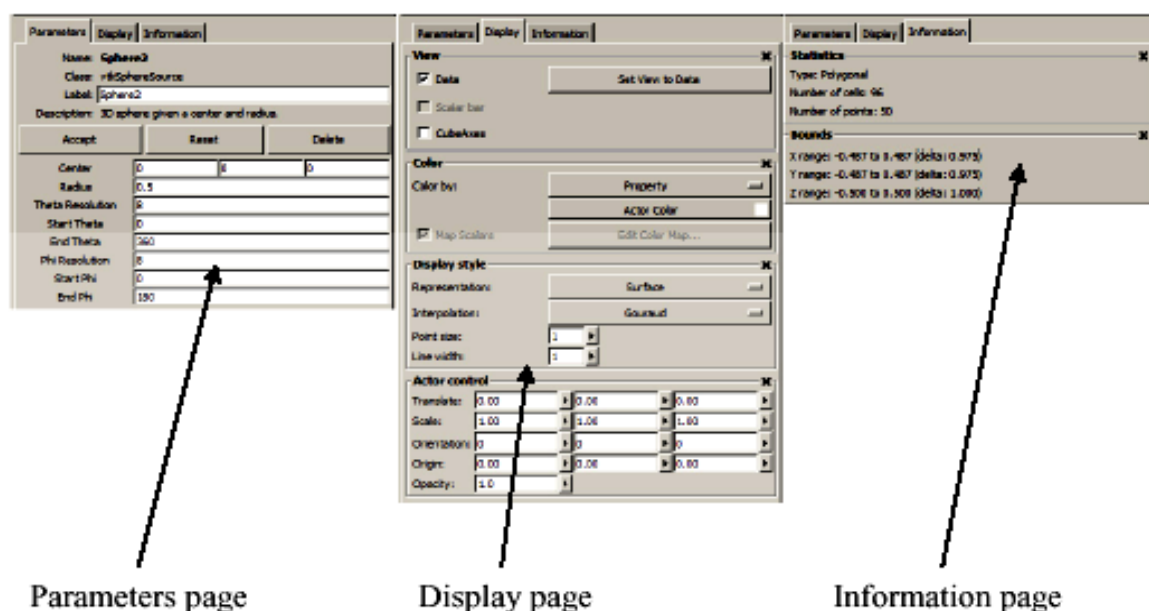
此时在属性条上的确认按钮将会突出显示成红色，标明次资源所创建的数据对象是废弃的。在这种情况下，它还未被真正创建。按下确认按钮



创建默认参数下的球体。你将会看到一个白色的球体现实在显示区。对于那些熟悉 VTK 的用户来说，这里就是所发生的细节。一个 vtk 球资源创建以后，这个对象的参数就在属性条

中显示给你。当你第一次创建它时，输出是空的，然后你需要按下确认按钮来激发更新函数来访问球体资源。当你在属性条处改变一个参数时，比较当前所显示的参数当然输出将会变得过时，那么只有更新才能显示当前结果。

但你往 ParaView 中添加数据时（通过加载，创建资源，或是过滤器），一系列的對象将会产生以控制数据的生成，数据的存储，数据的显示。ParaView 将这些对象分成组，组的名字在属性条中显示。在这个简单的



**Figure 3. Module Property Pages.**

例子中，就是 Sphere1。当你通过选择菜单或选择/导航窗口选择 Sphere1 时，所显示的属性条就包含了创建数据的资源对象的参数，所创建的数据对象的信息，和显示此数据对象的控件。例如，参数标签包含了针对球中心和半径的录入盒，用户可以由此而调整球资源的属性。信息标签告诉你通过球资源而创建的数据对象包含了带有 96 个单元和 50 个点的多边形数据。显示标签提供了与显示数据有关的对象的控件（对于 VTK 用户来说，这些就是影射器，角色，和属性对象），也允许你改变位置，

---

方位，颜色模式和其它的显示属性。

### 4. 3 改变成为丝形帧

你所看到的显示区域的图像应该是相似于左边所显示的。现在，我们将把一个实心的球体改变成为一个丝形显示，以更容易看到所隐含的多边形。要如此做，按下对应于左边面板中显示球体的属性条中的显示标签。在显示风格区，把表面显示改变为丝框显示。此时的图像就应该显示为相似于下图右边所示。

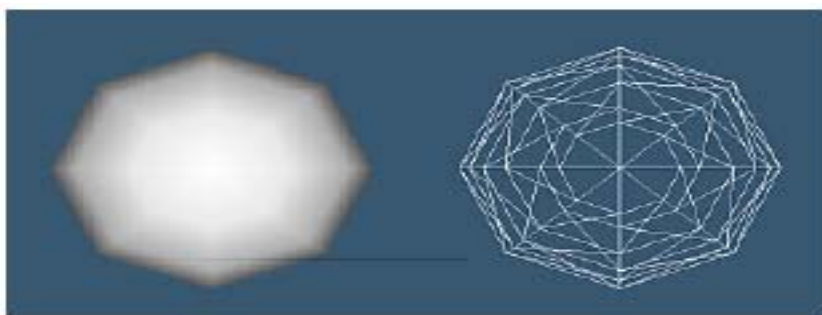


Figure 4. Solid and Wireframe Representations of a Sphere.

### 4. 4 改变决议

现在我们将要改变球体的决议。来到左部面板属性条的参数标签处。这里有两个决定性参数，Theta 参数和 Phi 参数，两者的默认值都为 8。从上面显示球的轮廓可以看出，由 8 段来定义一个球。

改变两者之值为  
改变成为红色，表  
户接口处已是过  
变后的结果。你将



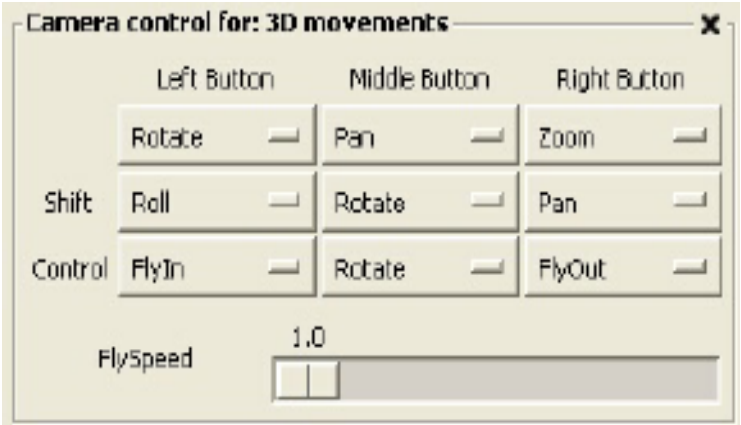
12。你将会看到确认按钮  
示显示对象相对参数在用  
时。按下确认按钮观看改  
会看到一个相似于上面所

显示的球体。这个球体比先前所显示球体包含了更多的多边形，并且提供了一个更近似于真实球体的模拟。注意，更改值并按下确认按钮并没有产生一个新的数据对象，它只是替换掉了当前所存在的对象。如果你已经改变了一些参数值并且确认按钮也突出显示为红色，但是你决定撤销此次更改，仅仅按下重置按钮。这将会恢复先前用于产生数据的值（或者是如果数据还未产生，就恢复默认值）。

### 4. 5 与球体交互

使用鼠标和键盘不同的组合，你可以控制显示区的对象和相机。鼠标绑定可以通过从查看菜单中选择三维察看属性选项来观察和更改，然后单击相机标签。你将会看到有两种类型的鼠标绑定：一个是用于二维运动，一个是用于三维运动。默认情况之下，相机处于三位模式之下（相机模式可以通过使用工具条而改变，看附录 D—相机控件的细节），在属性条的下面区域显示当前的绑定。默认的绑定显示于右边的图形之

中。上面一行标明在移动鼠标过程中，当你在显示区按下任一鼠标按键时会发生什么。第二行当你按下



行标明在移动鼠标过程中，当你在显示区按下任一鼠标按键时会发生什么。第二行当你按下

上特定的鼠标键时所要发生的动作，而第三行将会显示当按下 control 键及特定的鼠标键时将要发生的动作。通过使用下拉菜单你可以定制



---

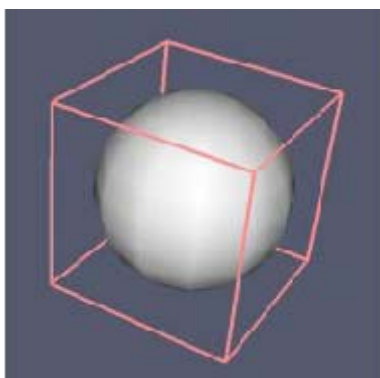
交互。

对于这个例子，试着使用左键来旋转场景。按下左键在显示区域移动鼠标将会导致相机在场景范围内旋转。相机旋转所围绕的点被认为是旋转中心。鼠标中间键能够用来转变场景，另外，右键能够用来放大和缩小场景。这三个操作是最为常用的。

## 4. 6 增加一个包围盒

在这一步中，我们准备增加一个过滤器。一个过滤器就是一个 VTK 对象，它处理一些输入的数据来产生输出。VTK 中包含了大量的过滤器，其中包括等高线过滤器，切平面过滤器，流线过滤器，以及低程  
度提取过滤器等。有一重要的一点需要提出就是过滤器并不改变它的输入。另外，它使用它的输入在输出中产生一个新的数据对象。

在菜单栏的过滤器菜单中选择轮廓项。这样做以后，你将会在左面板处看到轮廓属性条。就像资源操作一样，你需要点击确认按钮才能真正把这个过滤器添加到管道中。一旦你这样做以后，你将会看到一个包围盒



个包围盒围绕着球体。你同样会看到在选择窗口现在会出现两个选项：Sphere1 和 Outline0。两者都可看到，所以在任一左边都有一个睁开的眼。Outline0 对象是活动的数据对象，它突出显示为黄色。需要提出重要的一点是，当一个新的过滤器被创建时，它通常被应用为活动的数据对象。

使用 Outline0 的属性标签，通过点击颜色区域的角色颜色按钮来改



---

变线条的颜色。你也可以使用线条宽度滑动条来改变线条的粗度，这一点可以通过点击文字右边的小矩形框来进行，或者直接在编辑框中修改数字。

要访问 Sphere1 的属性，你可以在选择窗口中点击 Sphere1，或者在菜单栏中从选择菜单中选择该项。这将会使 Sphere1 成为当前的活动对象，并且其属性将会显示出来。使用显示标签将球改变成为表面显示。此时所显示的图像就将是如上图所示。

## 4. 7 缩小球体

我们现在将对球体应用另外一个过滤器。首先，确定 Sphere1 是选择窗口中突出显示的项，以便于我们能将添加的过滤器与之联系。现在从过滤器菜单中选择收缩过滤器。在属性条中，改变收缩因素为 0.75（默认情况下是 0.5）并按下确认按钮。你将会看到有几件事情发生。首先，选择窗口中新添加了一项叫 Shrink0 并且它是当前所选中的项。在显示区，由多边形所定义的球体已经收缩为原尺寸的 75%。实际上，原球体（带有原始尺寸的多边形）仍在那，只是现在看不到罢了。注意，挨着 Sphere1 的眼图标变为灰色标明这一项现在看不到。当添加了收缩过滤器后，基于输入数据而产生的新的数据对象（球体）就以产生。新的对象（收缩过滤器的输出）此时可见而原始输入不可见。

大部分的过滤器当它们被创建时都会改变输入数据对象的可见性。其中的两个例外就是轮廓过滤器（轮廓和轮廓角），因为假定使用这些过滤器用户想看到原始数据的同时看到过滤器的输出。


在这个例子中，你已经对一个数据对象（球体）应用了两个过滤器。按照 VTK 的术语，你已经创建了一个分支管道。你可以在导航窗口中看到这一点。要想从选择窗口改变到导航窗口，单击挨着选择窗口标题的  图标。你会看到标题更改为 Navigation Windows，并且在这个窗口中你会看到：



Figure 5. Navigation window showing a pipeline with one source and one filter.

暗灰色的项目为当前所选择的数据对象。其它所显示的蓝色的项目表明你可以单击这些数据对象来选择它。单击 Sphere1 项并且导航窗口显示的将会改变为：

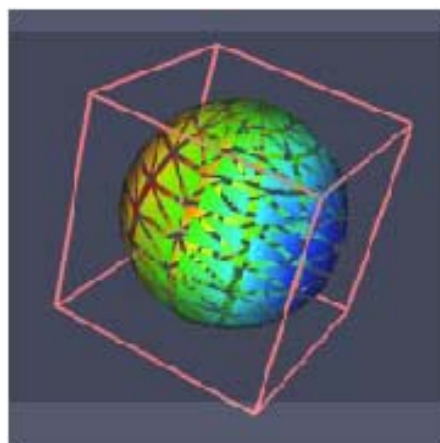


Figure 6. Navigation window showing a branching pipeline.

这是一个关于隐含型 VTK 管道的简单展示。我们已经创建了一个带有输出 `vtkPolyData` 的 `vtkSphereSource` 资源，并且作为了 `vtkOutlineFilter` 和 `vtkShrinkFilter` 的输入。

## 4. 8 使用法线来填色

在选择/导航窗口鲜红选择 Shrink0 项，访问该项目的属性条。在显示标签上，通过颜色区域的选项改变颜色为 Point Normals (3)。你现在应该看到一个相似于右边所示的图像。颜色是源自通过一个基



---

于数据的一些属性而形成的对应表而形成，而不是使用对象颜色的特定的颜色形成。在这种情况下，我们已经挑选了使用法线的 X 值来挑选介于蓝和红之间的颜色。

你可以通过使用观察区域的编辑颜色映射按钮来改变当前的颜色映射。这将会把左边面板显示的属性条改变成能够用于控制颜色映射，和控制标量栏的各种不同属性。在颜色映射区，你可以改变颜色映射的两个末点。一个 HSV 的中间插补策略服务于中间值。

第一个应用实例到此为止。现在你可以通过选择文件，退出或者关闭窗口来退出 ParaView。默认情况之下，ParaView 将会跳出一个对话框询问是否真正要退出。你可以选择 “Do not show this dialog anymore” 的选中框来关闭这项特色。如果你这样做，并且想以后重新使用这个推出对话框，你可以到查看，应用程序设置中，开启 “Confirm on exit”。

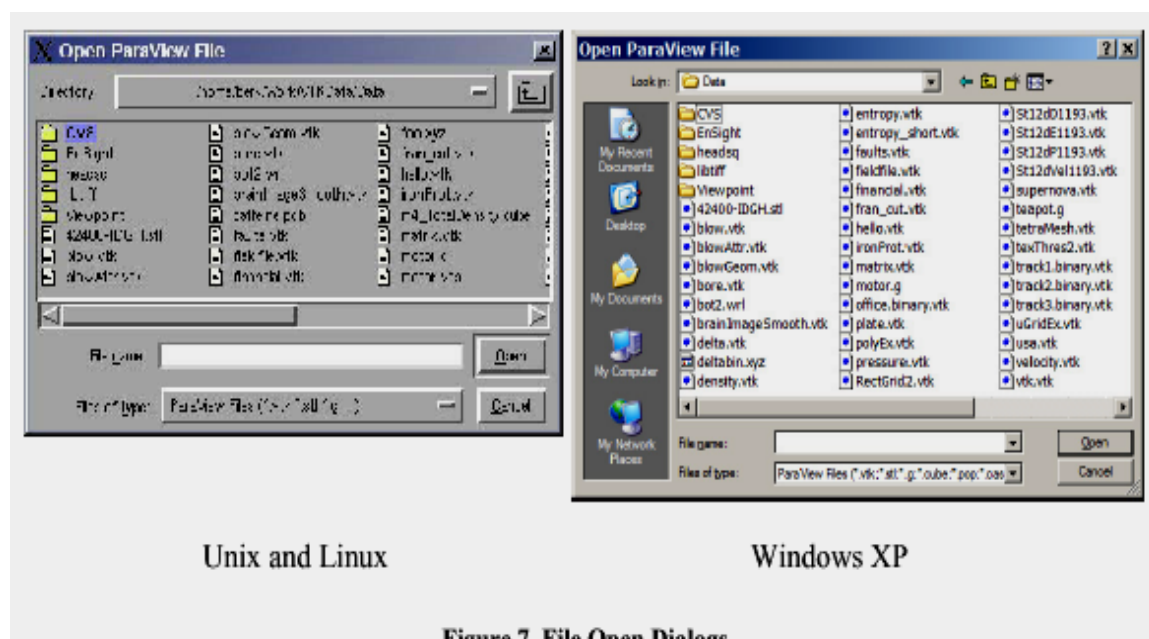
## 5 实例 II 一流线可视化

这个例子将会介绍更多的基本概念，其中包括加载数据文件，与三维窗口部件进行交互和保存数据文件。如果你对 ParaView 不熟悉的话，我们建议你在通过这个例子之前先读一下第一个实例。

### 5.1 加载数据文件

本实例假定在遵循实例步骤之前你还没有采取任何行动。不然，所创建的模块名字将会改变。关于怎样开始 ParaView，请看实例 I。

要加载本实例中所使用的数据文件，从菜单栏中选择“File, Open Data”。这将会开始运行如下所示的文件选择对话框的文件之一。



最初，文件选择对话框显示所有 ParaView 所承认的扩展文件。你可以通过使用“Files of type”菜单选择不同的过滤器来限制显示的文件。参考附录 A—文件格式来得到关于 ParaView 所支持的更多的文件格式信息。

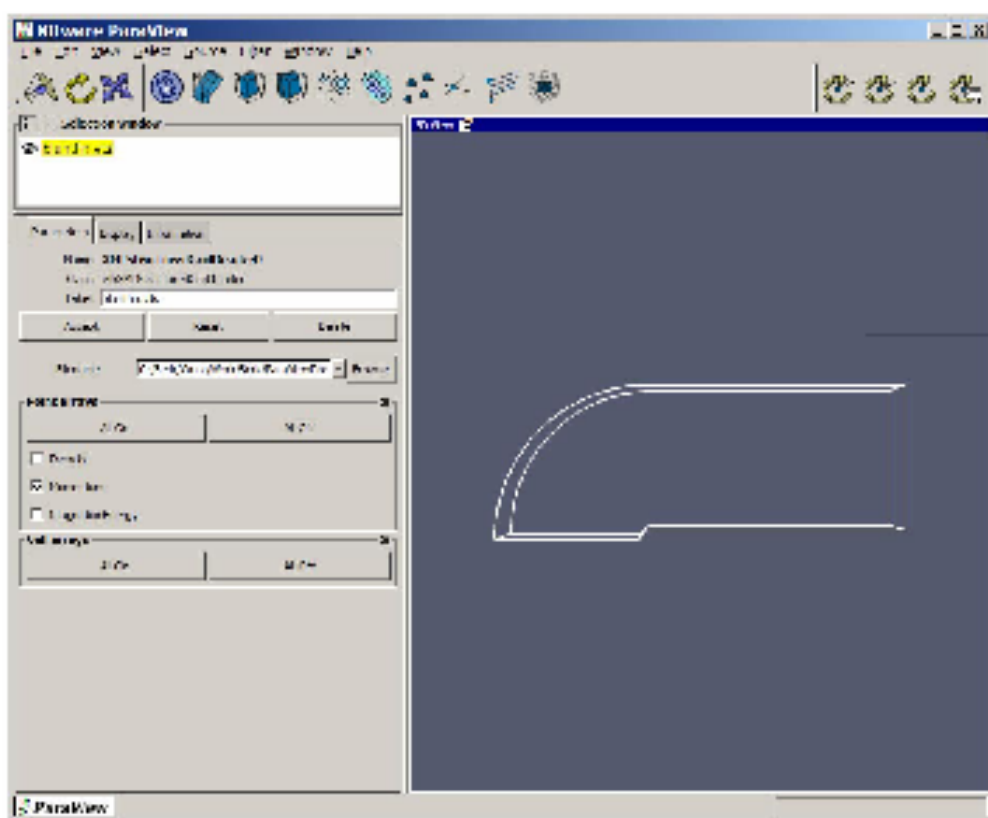
导视到实例数据文件所位于的文件夹。实例数据文件与文档一起分



布。它们分布在实例数据文件夹。下一步，选择 bluntfin.pvs（如果文件没有列出，确保文档安装正确，并且通过选择“Files of type”菜单中的 ParaView 文件文件选择对话框能够显示所有的 ParaView 文件类型）并且点击打开。左边面板将会显示成为左

边所示。到此时为止，ParaView 读了关于数据文件的信息，但是并未

真正独到任何数据。这就允许 reader 在文件被读入之前，更改“reader”的某些属性。正如图片中所示，此时属性框中包含了文件中可用的所有数据列（变量，属性）。在这种情况下，文件包含了两个标量变量，分别为密度和能量停滞，还有一个矢量变量称作动力。在默认情况下，如果你点击确认，那么所有的数列都被选择并将会被读入。然而，既然你在这个实例中仅需要动力矢量，你可以不选择那两个标量以缩短加载时间和降低内存消耗。假如一个“reader”没有用户可使用的属性，它将会立即加载数据文件。在没有选择密度和能量停滞，点击确认。此时 ParaView 窗口将会如下所示：



**Figure 8. ParaWindow after opening a structured grid data file.**

你刚才所下载的文件包含了一个结构化的网格数据，它通过一个从平盘上而产生的钝翅而显示一个在平盘上面的超音速流。它是通过一

个由 NASA (<http://www.nas.nasa.gov/Research/Datasets/datasets.html>) 所分布的 PLOT3D 文件而转化的。你将会看到近显示了网格的一个轮廓帧。ParaView 把结构化的数据（矩形边界和曲线网格）显示为轮廓，把非结构化的数据（多边形数据和非结构化的网格）显示成为表面。关于 ParaView 所支持的数据类型的更多的信息，参看附录 B—数据类型。

## 5. 2 提取子网格

下一步，我们将使用提取过滤器从结构化网格中提取两个二维的子网格。单击工具栏上的提取图标创建第一个过滤器。在左部面板所显示的 VOI（Volume of Interest）属性中，通过使用滑动条或测定入口盒把 I 的最大和最小值设为 18。点击确认，输出就应该如下所示：

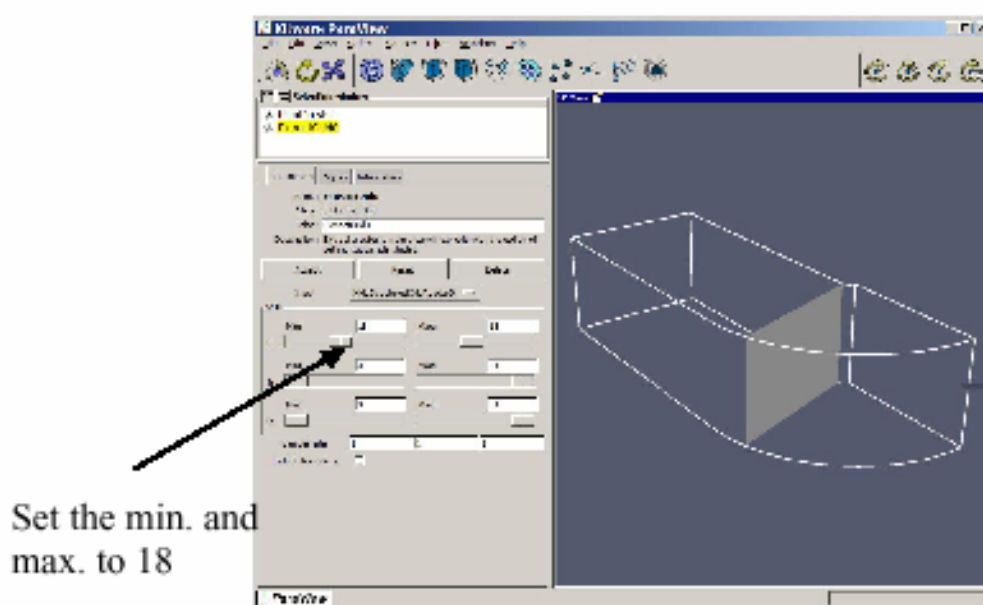


Figure 9. First sub-grid.

提取是一个过滤器，它要么从一个结构化的网格数据中提取一部分，

要么使它子例子化。被提取出来的一部分就被称作 the Volume Of interest 或者 VOI.抽取概率是由“抽取概率”参数来决定的。注意上面的屏幕是在改变了照相机的位置以后而产生的，这一点使用了上个例子中所提到的三维交互模式。要改变这个新平面的颜色映射，选择显示页并且通过菜单“Point Momentum (3)”来选择颜色。接下来，我们将对 fin 表面进行提取。首先，确定三维结构化网格是当前活动对象（点击选择窗口中的 bluntfin.vts）。现在通过点击工具栏上的提取过滤器图标来创建另外一个过滤器。接下来，设置 J 的最大值和最小值为 0（在 VOI 属性里），两种设置方式同上。单击确认。ParaView 此时应该如下所示：

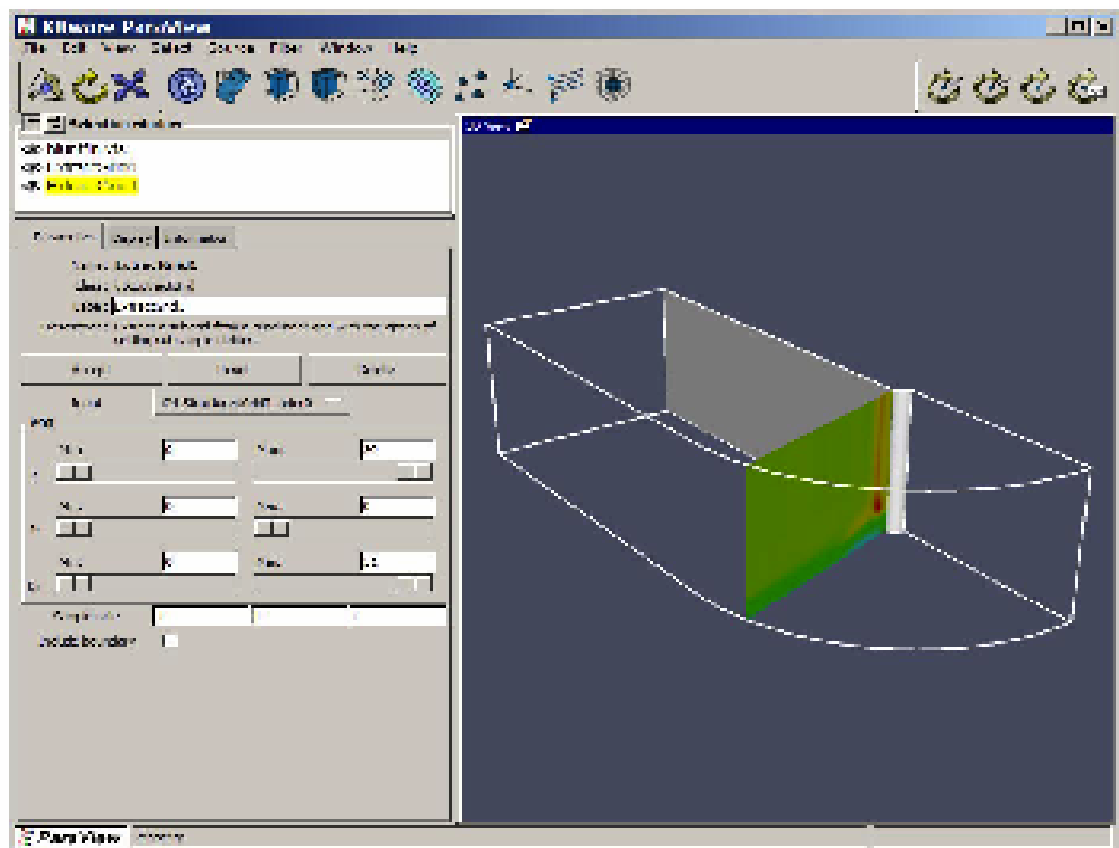



Figure 10. Second sub-grid.



## 5.3 创建流线

 接下来，我们添加一些流线。首先，确保三维结构化网格处于活动状态（在选择窗口单击 **blunffin.vts**）。然后，通过点击工具栏上的流线描绘来创建一个流线描绘过滤器。流线描绘过滤器比先前所用的过滤器有着更为复杂的属性。其中最为重要的是种子的属性，其中包括两个成分。首先一个显示在属性页并且允许你文本式的更改种子设置。第二个显示在显示区并且允许你交互式地更改种子的设置。要想在显示区域看到点部件，你可以通过点击选择窗口中挨着 **ExtractGrid0** 的眼图标来开启它的可见性来完成。

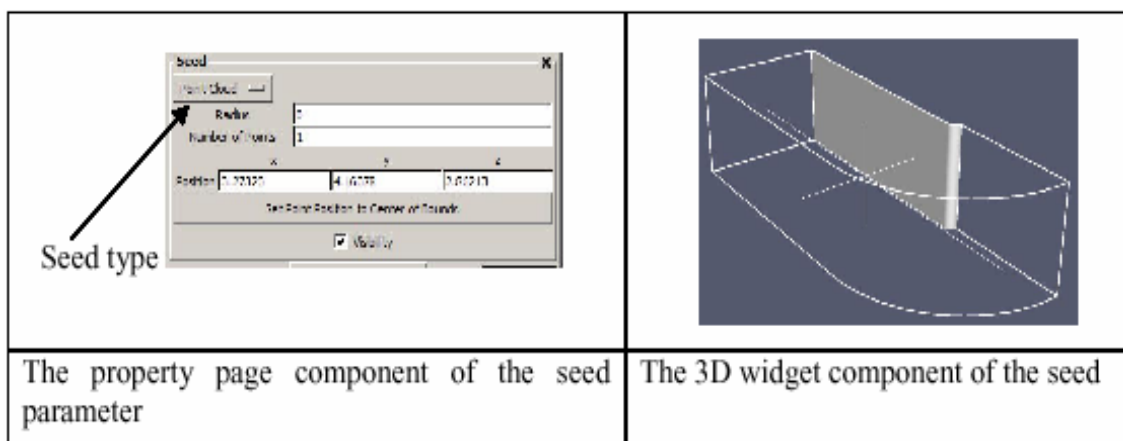
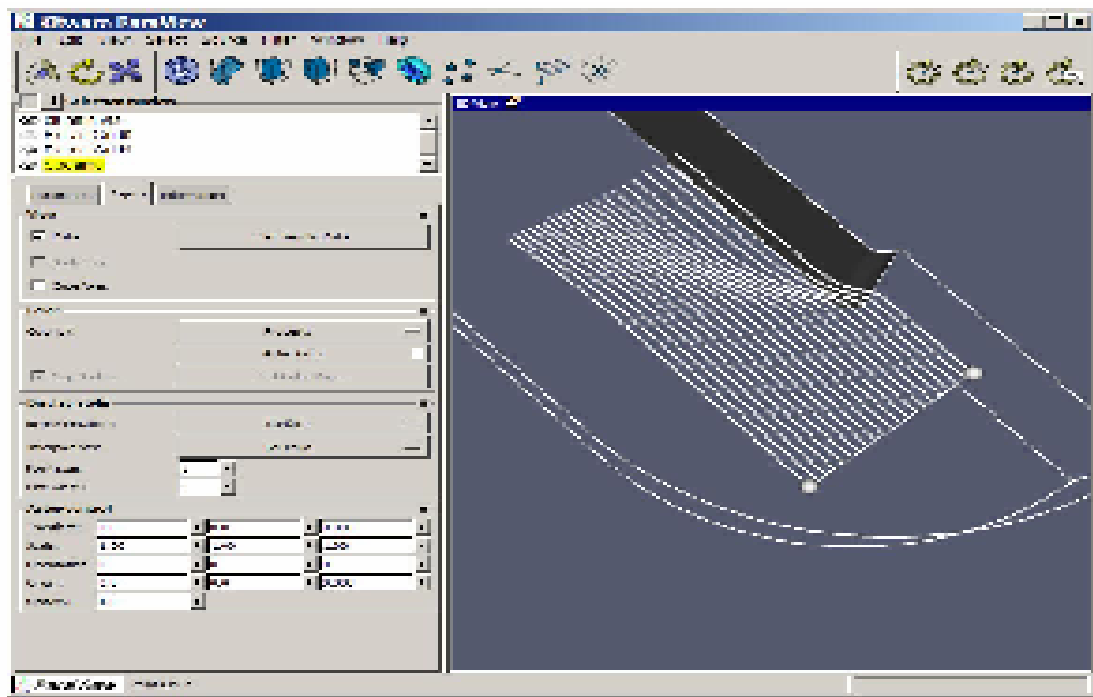


Figure 11. Components of the seed parameter.

ParaView 支持这些操作者中的一部分，一次允许用户很容易的改变参数。关于在 ParaView 中所使用的一系列的三维小部件，参看附录 C—三维小部件。你会首先使用一条线来从流线中提取种子。从种子属性帧的下拉菜单中选择线条作为种子类型。接下来，通过变动入口盒设置点一为  $(-3.59, 0.33, 2.86)$ ，点二为  $(-3.59, 5.48, 2.86)$ 。你也可以通过点击线的每条边的圆球并进行拖动来改变末点的位置。接着，设置最大传播长度为 10。单击确认。此时的输出应该如下图所示：





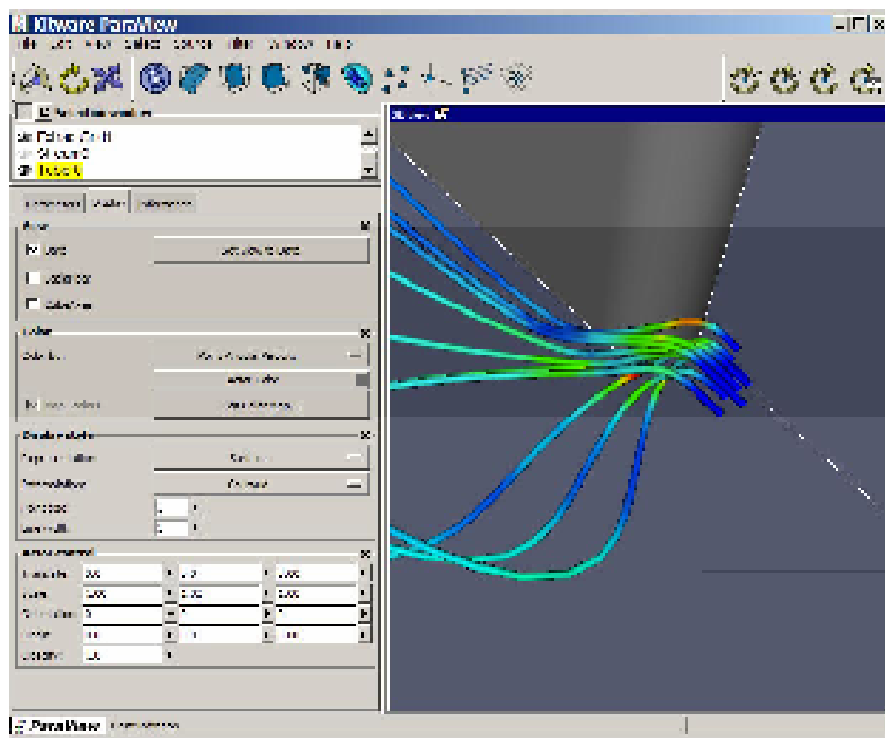
**Figure 12. Streamlines created with a line seed.**

接下来，我们将种子移到更有兴趣的地方。首先更改种子类型为 Point Cloud；该半径为 0.3，点的数目为 10。然后，更改云点中心位置大约在  $(-0.54, 0.28, 0.55)$ 。这个点接近于位于平面和部件交界处上的停滞点。单击确认。云点种子将会产生 10 个点，它们任意分布在半径为 0.3 的球体周围，中心在部件点的位置。要想更为详细地观察交接区域的结构，可以把相机移到云点中心更为接近的地方。ParaView 提供了一些设定相机位置比较容易的工具。关于这方面更为详细的讲解，参看附录 D—相机控件。

## 5. 4 增加流管

在三维情况之下，没有阴影的多边线不是最好的流线可视方法。增加一些粗度（和阴影）可以更为容易的理解流线领域的结构。这方面可以使用流管和带过滤器。流管过滤器在输入线周围产生流管而流带

过滤器由输入线产生导向流带。在本实例中，我们将会使用流管过滤器。首先确定流线过滤器的输出处于活动状态。（在选择窗口中选择 Stream0 或者从选择菜单中选择 Stream0）。接下来，从过滤器菜单中选择“Tube”来创建一个流管过滤器。设置“Number of Sides”参数为 8，半径为 0.03。单击确认。你可以通过选择属性页的显示标签并且从菜单中选择颜色来更改颜色映射。此时 ParaView 窗口应为如下所示：



**Figure 13. Tubed streamlines.**

你也许已经注意到了过滤器菜单的内容并不一直都一样。这是因为过滤器菜单随文本变化，并且只显示适用于当前数据对象的过滤器。例如，如果当前对象是结构化网格类型，那么流管过滤器就不在菜单之中，因为流管过滤器只适用于多边形边。类似的，如果当前对象不包括任何的三维向量，则流线过滤器也不可用。一旦创建或选择了一个

---

新数据对象，ParaView 就会通过比较每一过滤器的要求和当前对象的属性并且仅添加可用的过滤器来更改过滤器菜单的内容。

## 5. 5 保存流管

因为本实例中所使用的例子数据文件比较小，所以流线计算花费的时间很少并且能够很顺利的进行反复。然而，随着数据文件的扩大，这种计算可能会花费很长的时间。在这种情况下，将计算的结果保存起来可能有用，以便以后能够在此保存和在三维情况之下查看。要保存流管的输出，从选择窗口（或从选择菜单）之中选择对象 Object0 并且选择文件保存数据。这将会产生一个保存文件的选择窗口，这一点非常类似于打开文件对话框。可以把路径更改到你所想文件保存的地方。在这种情况下，你可以把数据对象写成三种不同的类型：ParaView 数据类型 (\*.pvd)，VTK 多边形数据形式 (\*.pvd)，遗传 VTK 形式 (\*.vtk)。

第二个实例到此为止，现在你可以要么选择文件，退出，要么直接关闭窗口来推出 ParaView。

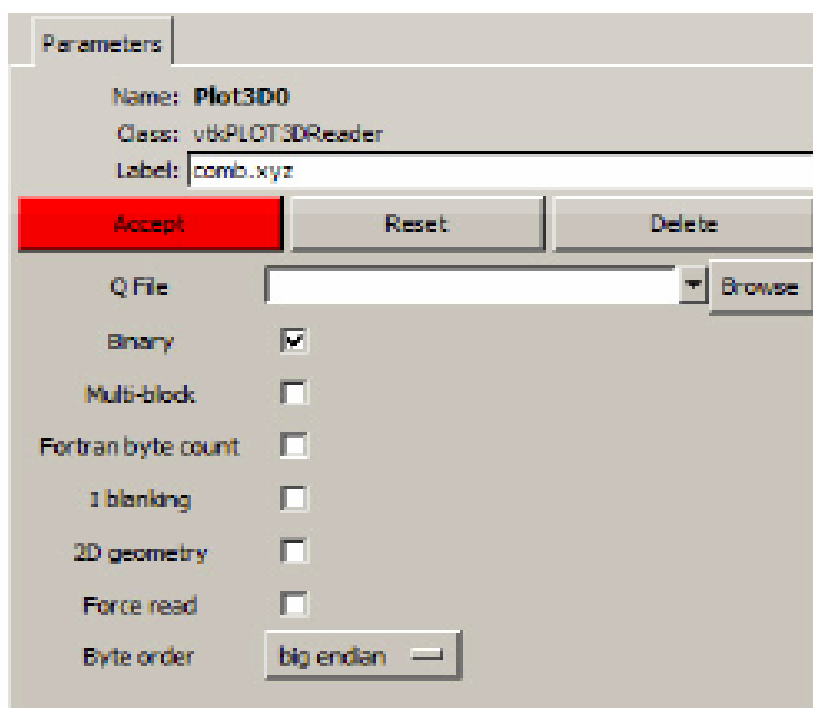
## 6 实例III—动画等高面

这个实例将会介绍 ParaView 的动画接口。它将会展示一些 ParaView 的过滤器的用法，包括登高面，修剪和剪切过滤器，并且演示怎么使用动画编辑器对它们进行控制。如果你对 ParaView 不熟悉的话，我们建议你在通读本实例之前先读一下前面的两个例子。

## 6. 1 加载数据文件

本实例假定在原定步骤之前你还没采取任何的行动。不然所创建的模块名字也学会有所变化。关于怎样开始运行 ParaView 可参看实例 I。

在这个例子中，我们将会使用一个 PLOT3D 文件。关于 ParaView 所支持的文件类型的更多的信息，可参看附录 B—数据类型。PLOT3D 格式的文件把几何结构（结构化的网格）和属性分存于不同的文件之中。首先通过选择文件，打开菜单打开几何数据文件，导航到实例文件所位于的文件夹处（实例文件与文档结合分布；它们分布在实例数据文件中），从数据选择对话框中选择 comb.xyz，单击打开。现在，你应该看到一个关于 PLOT3D “reader” 的参数页如下：



**Figure 14. PLOT3D reader parameters.**

到现在为止，参数中的一些项目当它们需要时已经被介绍了。下面是关于这些项目的更为详细的介绍：

- 
- 名字：ParaView 为每一 reader，资源和过滤器分配了一个独一无二的名字。这个名字被 ParaView 的成分包括选择菜单，选择窗口（通过从查看，应用程序设置菜单中选择显示资源名字）。名字不可以被改动。
  - 类：VTK 中隐含对象的类名。类的名字仅用于提供信息。
  - 标记：在默认情况之下，ParaView 使用这个名字在浏览窗口（选择/导航窗口）之中来显示模块（readers，资源，过滤器）。尽管 ParaView 初始时设定了一个默认值，但用户可以改变标记以更容易的放置模块。
  - 确认按钮：当改变多个参数时为了避免不必要的过滤器操作，ParaView 并不在一个参数改变时应用一个过滤器。而是当一个改变发生时，确认按钮突出显示为红色以提醒用户一个或多个参数已经发生了改变。要想应用这些改变并且执行操作，点击确认。
  - 重置按钮：要想重置模块的状态到先前操作时的情况，使用重置按钮。
  - 删除按钮：要移除一个模块，使用删除按钮。注意当一个模块的输出正被另外一个模块使用时，删除按钮处于无效状态。为了删除这样的模块，你不得不删除所有使用其输出的过滤器。

其余的参数是特定于 PLOT3D reader 的。要得到关于这些参数更为详细的信息，你可以参考帮助。如果工具提示可用的话（查看，应用程序设置，现实工具提示），当鼠标移到一个参数旁边几秒后，一个关于该参数的描述的小窗口将会直接显示出来。在这种情况下，你不必更改任何的 PLOT3D 参数。点击确认。接下来，改变标签到 Combustor 数据（确保在你点击新标签后按下进入，不然改变将不能生效）。通过

---

指定更多有意义的标签，你可以使在选择窗口中寻找一个模块更为容易；尤其是当存在很多模块时。

你也许已经注意到了，除了 Q 文件以外其它的 PLOT3D 参数都是不可编辑的。PLOT3D reader 与其它的 reader 模块不同。当它第一次创建时，它将会允许你设置它的参数直到符合了文件本身的参数。这一点是通过使用已给的参数读文件的一部分来完成的。直到文件能够被读入，确认按钮一直保持红色。这样做可以提醒用户参数还不正确。一旦文件被读入，参数就不需要进行修改了。然后，它们变为不可编辑状态。这些步骤是必须的，不想其它 ParaView 所支持的文件格式，PLOT3D 文件不包含任何用于写它们的参数的信息。

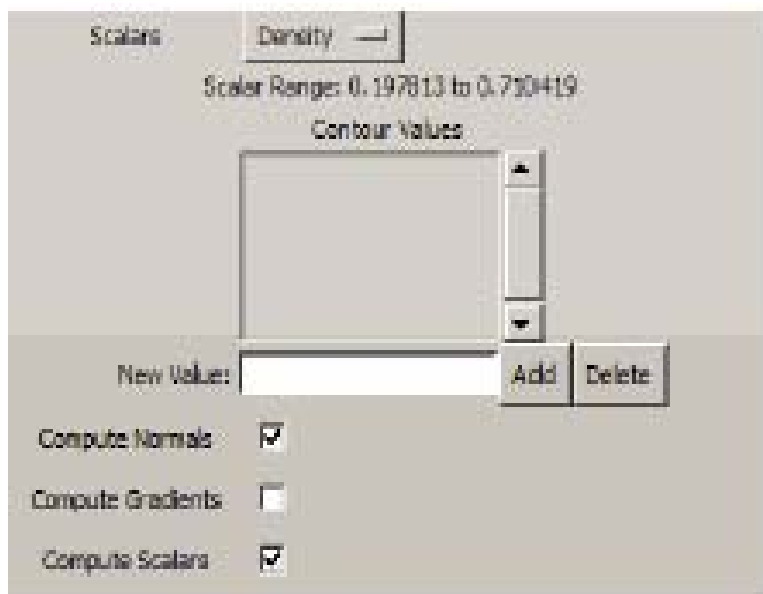
一旦文件被加载，在工具栏中的部分按钮将变为可用，而其它的则变为不可用。这是因为，就像过滤器菜单一样，工具栏按钮也是文本敏感型。到目前为止，你仅仅加载了几何文件。所创建的数据对象不包含任何的数据列（标量，矢量等）。因此，像需要数据列的过滤器，比如数列计算器，偏差，极限，等高线等都是不可用的。要想加载这些属性，单击挨着 Q 文件参数的浏览器按钮，从文件选择对话框中选择 comb. q 并点击打开。

## 6. 2 创建一个等值面



接下来，我们将会使用等值面过滤器来提取一个密度等值面。你可以要么选择工具栏上的等高面过滤器，要么选择过滤器，登高面来创建一个带有一个空系列的等高面值的过滤器。此时的参数应该如

下图所示：



这里，第一个参数部件允许用户选择用以创建等值面而使用的数列。任一个成分数列都能用于创建等值面。如果你点击选择部件，你将会看到可用的数列是密度（density）和

SgnationEnergy。下面显示的是当前所选择的数列的取值范围。你可以通过使用登高面值部件来添加等值面值。要从中移除一个值，选中它并点击删除即可。下面，点击确认。也可以改变等高面过滤器的标签到活动等高面。此时的 ParaView 窗口应该是如下所示：

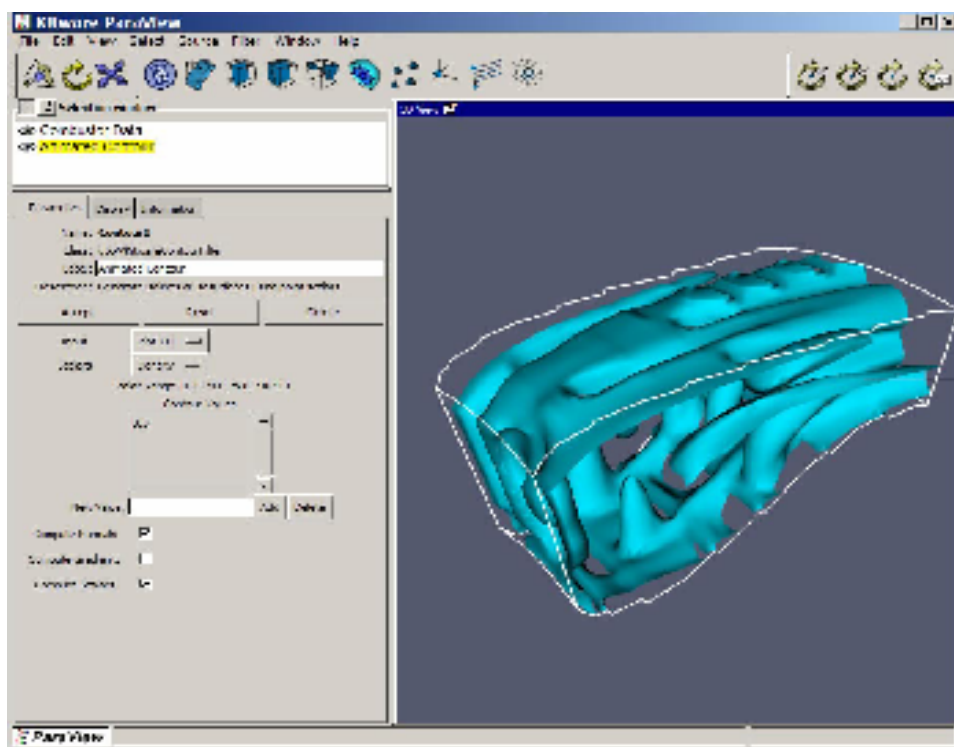


Figure 15. First contour.

## 6. 3 剪切等值面



因为所显示的等值面覆盖了所有网格的大部分，所以很难看到范围中心的一些细节。你可以使用剪切过滤器来移除等值面的一部分。首先确保选中了等高面过滤器（活动的过滤器）。通过点击工具栏上的剪切按钮和选择过滤器，剪切来创建一个剪切过滤器。平面的方位和中心可以进行修改，可以通过与显示区域中的三维部件进行交互或调整合适的入口盒来实现。参看附录 C—三维部件可以得到大部分常用的三维部件的描述。在这个例子中，平面默认的参数很好。单击确认。你将会看到剪切过得等值面替换掉了整个的等值面。这是因为当一个剪切过滤器创建以后，它的输入的可见性被 ParaView 自动进行更换。改变剪切过滤器的标签到剪切登高面。接下来，通过选择显示标签并且更改 Color By 为 Point Momentum (3) 而改变表面的颜色映射。此时 ParaView 窗口将会显示如下图所示：

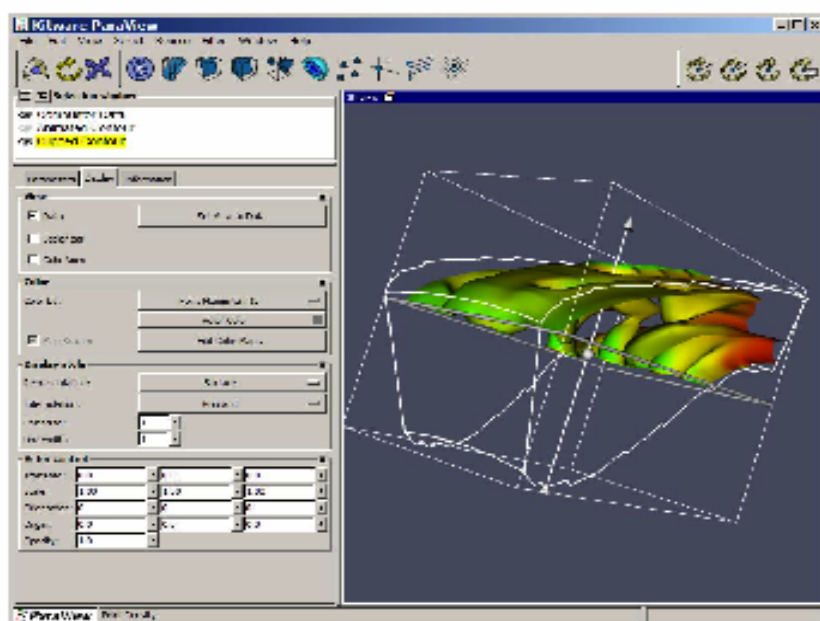


Figure 16. Clipped contour (colored by the first component of Momentum).

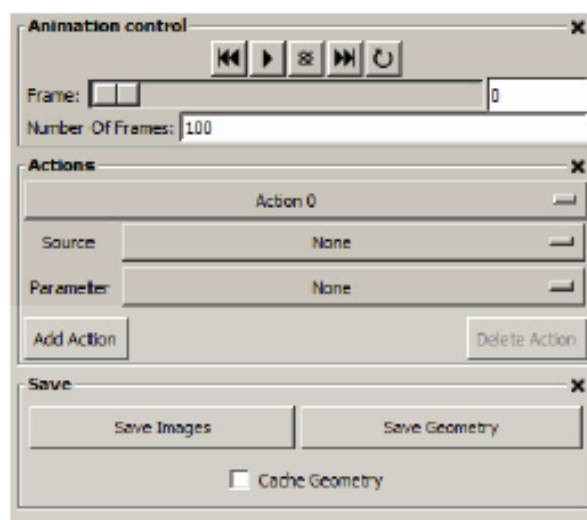


你可以通过切换到参数页并关闭剪切功能的可见性来关闭平面部件的可见性。

## 6. 4 动画等值面

在这一节里，我们将展示怎样使用 ParaView 的动画编辑器来在等值面值的范围之内进行动画。要显示动画编辑器，选择查看，动画。此时左边面板将会如有图所示。

上层按钮就像一个 VCR 那样工作——第一个按钮可以用来返回到开始，下面的按钮进行演示，中间的方块如果开始演示可以停止它，双重向右的箭头将会到达末尾，最后一个按钮是一个绳针



按钮可以用来进行循环。其中帧进入盒的数目确定动画中总的帧数，而帧滑动条则决定了当前所演示的帧。

当按下播放按钮时，通过添加将要演示的动作来创建动画。通过首先选择 **Contour0** 作为资源，等高线值作为参数来创建你的第一个动作。注意，资源选择菜单列出了可用的使用它们独一无二名字的资源，而不是由用户所指定的标签。如果你不能记起资源的独自的名字，你可以在资源的参数条的顶部找到它。接下来设置开始值为 0.3，结束值为 0.6。这里就是你刚才所创建的动作：当第一个矿建被显示时设置 **Contour0** 过滤器的等高线值参数（当前，从动画编辑器中仅能改变一

---

个等值面值) 为 0.3, 并且当最后一个帧显示时改为 0.6; 对于其间的任意一帧, 线性的插入介于 0.3 和 0.6 之间的值。这是一个非常简单的关键帧动画模型, 第一个和最后一帧作为关键帧而只允许中间线形插值。你可以通过使用增加动作按钮并且借助资源、参数、开始值和结束值来描述动作, 来任你添加尽可能多的项目。接下来, 把帧的数目减少为 20。

要开始动画, 按下 VCR 控制面板上的播放按钮。你将会看到动画播放的相对有点慢。这是因为, 每一帧都因为运行了等高面过滤器而创建了一个新的等值面。这是一个相对扩展了的操作。要想避免这样做, 可以在等高面过滤器第一次运行时缓存显示每一等值面的多边形数据。在动画编辑器页的底部通过选中缓存几何项来打开缓存。当你第一次运行动画时, 它可能仍是慢的。可是当你第二次运行动画时, 你将会看到快了很多。使用几何缓存的缺点是缓存把每一步的几何信息都保存在了内存中。当步骤较多或几何信息很庞大时, ParaView 可能会把内存用空。要移去几何信息, 可以使缓存几何无效。

## 6. 5 删除等高和剪切过滤器

在到另外一个动画之前, 我们将会展示怎样删除 ParaView 的模块。首先, 通过选择查看, 资源显示左部面板的资源页。接下来, 在选择窗口中选中动画的等高线模型。你将会注意到删除按钮不可用。

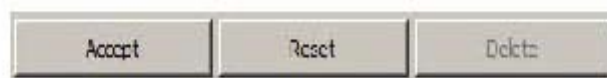


Figure 17. If the output of a source is used by another filter, that source can not be deleted.

这是因为等高线过滤器的输出被剪切过滤器使用着。你可以切换到导航窗口中来验证这一点。

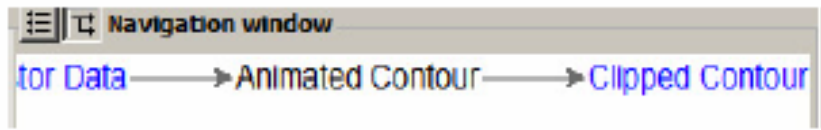


Figure 18. Navigation window showing part of the pipeline.

ParaView 不允许删除一个模块，如果它的输出正被另外一个模块使用。要删除等高线过滤器，首先选中剪切过滤器（Clipped Contour），删除它然后再删除等高线过滤器。也可以选择编辑，删除所有模块来删除所有的模块。然而，我们将会在这个例子中需要 reader 模块，所以不要使用“删除所有模块”。

## 6. 6 动画一个剪切面



在这一节中，我们将通过 combustor 数据结构并且把它移入到一个动画之中来创建一个剪切面。可以点击工具栏上的剪切按钮或选择过滤器，剪切来 0 创建一个剪切过滤器。改变标签到剪切平面，通过点击剪切功能属性的可见性来关闭三维部件的可见性，另外点击确认来接受所有其它的默认参数。ParaView 窗口将会如下图所示：

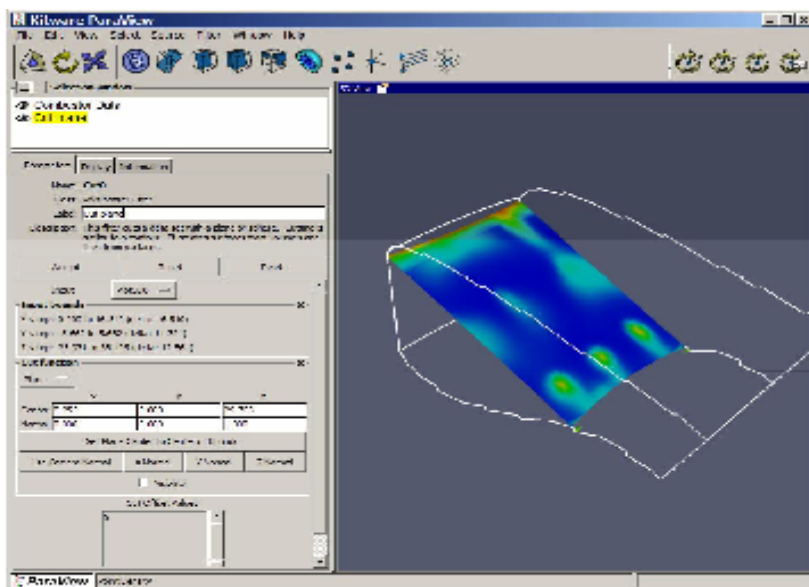


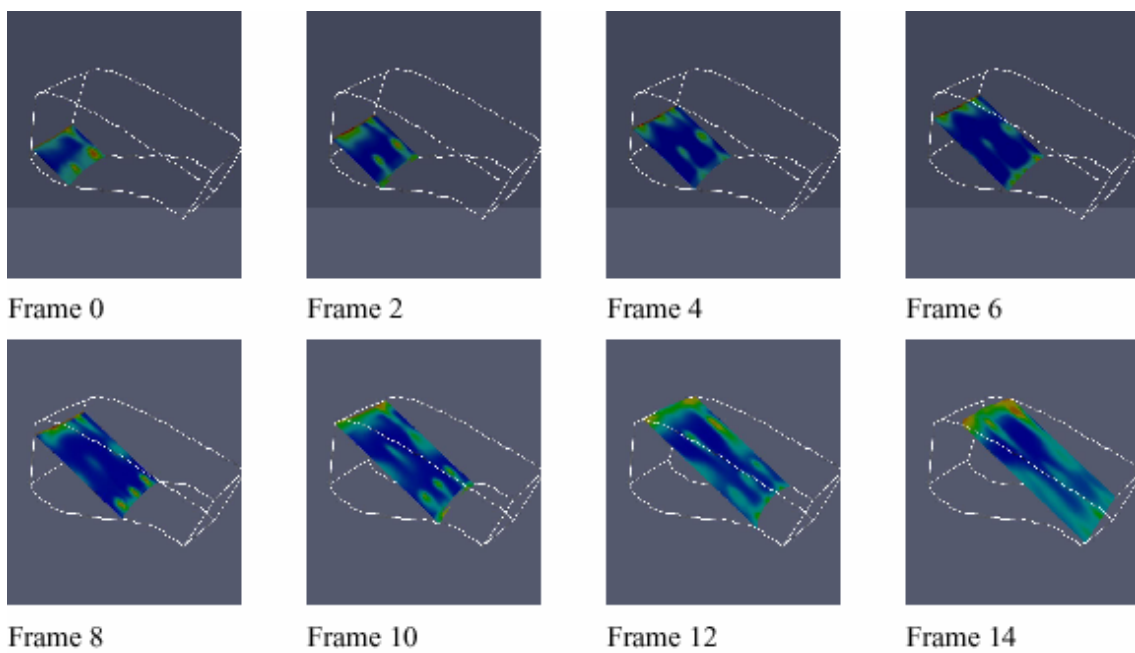
Figure 19. Cut-plane to be animated.

---

然后，通过选择查看，动画而切换到动画编辑器。你将会看到以前的动作已被删除。不论何时只要一个模块被删除，ParaView 就会寻找该模块所使用的所有动作。所有符合的动作都被删除。通过选择 Cut0 作为资源，Cut 值作为参数，-5 做为开始值，5 做为结束值来添加一个新动作。确保缓存几何打开，按下 CVR 控件中的播放按钮。当下次播放动画时，它将会运行的更快。事实上，在一个快速的计算机上重放的速度也许会太快。

要添加帧与帧之间的延迟，你可以使用脚本编辑器。脚本编辑器允许你再每一帧处写一个被执行的 Tcl 脚本。这主要是针对高级用户，在这里将不做过多的讲述。通过选择参数下拉框菜单中的脚本来打开脚本编辑器。在脚本的末尾添加一行“after 100”，即可在每一帧中间加上一个 100ms 的延迟。当下次你播放动画时，你会注意到运行减慢。

ParaView 可以保存每一帧为图像。这些图像可以以后搜集起来，使用普通的可用工具来创建一个动画文件，比如 AVI 或 MPEG 等。要注销这样的图像，首先准备动画；然后确保显示区没被其它的窗口所覆盖，然后点击保存图像。ParaView 将会运行动画并且为每一帧保存一个图像文件。本节中练习中所有帧的一部分将会如下所示：



第三个实例到此为止。现在你可以要么选择文件、退出，要么关闭窗口来退出 **ParaView**。

---

## 附录 A—文件格式

在默认情况下，ParaView 支持下列文件类型：

- **ParaView 文件：**这是 ParaView 所默认的文件形式。有这种 reader 所创建的数据对象也许是由 ParaView 所支持的类型中的任一种（多边形，图像，矩形，结构化网格或非结构化的）。文件的扩展名是.pvd。这种格式的文件支持控件分区数据和多块数据。
- **VTK 文件：**XML 基于由 VTK 所使用的文件格式。有这种 reader 所创建的数据对象也许是由 ParaView 所支持的任一种（多边形的，图像，矩形，结构化网格或非结构化的）。其文件扩展名是：对多边形数据.vtp, 对图像数据.vti, 对矩形数据.vtr, 对结构化网格.vts, 对非结构化网格.vtu.
- **并行（分区的）VTK 文件：**XML 基于由 VTK 所使用的文件格式。并行的 VTK 文件包括了关于数据空间分布的信息，并且可能指到多个 VTK 文件。有这种 reader 所创建的数据对象也许是 ParaView 所支持的任一种（多边形的，图像，矩形，结构化网格或非结构化的）。文件扩展名是：对于多边形.pvtp, 对图像数据.pvti, 对矩形数据.pvtr, 对结构化网格.pvts, 对非结构化网格.pvtu.
- **继承的 VTK 文件：**这是遗传下来的 VTK 文件格式（在 VTK4.2 之前，尽管仍然支持）。所有类型的数据都以同样的扩展名而被保存。这种类型的文件将会有有一个.vtk 的扩展名。有这种 reader 所创建的数据对象也许是 ParaView 所支持的任一类型（多边形的，图像，矩形，结构化网格或非结构化的）。

---

- **并行的继承文件：**继承的 VTK 文件格式（尽管仍支持，在 VTK4.2 之前）所有的数据类型都按同一扩展名而保存。有这种 reader 所创建的数据对象可能是 ParaView 所支持的任一种（多边形的，图像，矩形，结构化网格或非结构化的）。

- **ensight 文件：**这是由 CEI 的 EnSight 所使用的文件格式。（<http://www.ensight.com>）。支持 ASCII 和二进制 EnSight6 和 EnSight Gold 文件格式。这些文件的扩展名是：.case. 有这种 reader 所创建的数据对象也许是由 ParaView 所支持的任何一种类型（多边形的，图像，矩形，结构化网格或非结构化的）。这种格式同样支持多部分（参看关于多部分和多块数据的 ParaView 用户指南信息）。

- **PLOT3D 文件：**这种文件格式初始时由 NASA 所开发的测绘包所使用。ParaView 假定的默认的扩展名是：对几何文件.xyz，对方案文件.q。它仅支持结构化网格数据格式（二进制，ASCII, 单一的或多块的）。

- **Stereo Lithography：**ParaView 可以读入二进制的或 Stereo Lithography 文件。这些文件的扩展名是.stl. 这种格式仅支持多边形数据。

- **BYU 文件：**ParaView 可以读 MOVIE.BYU 文件。这种文件的扩展名是：.g。这种格式仅支持多边形数据。

- **POP Ocean 文件：**在能源部的赞助下由 Los Alamos National Laboratory 所开发的 Parallel Ocean Program 软件所支持的文件。（<http://www.acl.lanl.gov/climate/models/pop/>）。这些文件的扩展名是.pop。这种文件格式仅支持结构化网格数据。

---

• **Protein Data Bank 文件：**由 Protein Data Bank 所使用的文件格式，一种实验决定的大生物分子三维结构的档案文件 (<http://www.rcsb.org/pdb/>)。这种文件的扩展名为：`.pdb`。这种 PDB 的 reader 能产生多边形的输出数据。

• **XMol 文件：**Minnesota Supercomputer Center（明尼苏达超大型计算机中心）的 XMol 文件格式。XYZ 是展示分子的一种简单文件格式。它描述分子和结合物，但分子上没有任何的值。这些文件的扩展名为：`.xyz`。XMol reader 能够产生多边形的输出。

• **XDMF 文件：**这种可扩展的数据模型和格式是一种活动、常用的数据中心，是在应用程序模块之间传递值和元数据的一种标准方式 (<http://www.arl.hpc.mil/ice/>)。这些文件的扩展名为：`.xmf`。这种格式支持矩形和非结构化网格。

对于其它格式的支持可以通过写一个简单的 XML 配置文件（如果 VTK reader 已经存在的话），或者通过 C++ 语言实现 VTK reader。细节可参考 ParaView 的用户使用指南。



---

## 附录 B—数据类型

ParaView 所支持的数据类型有：

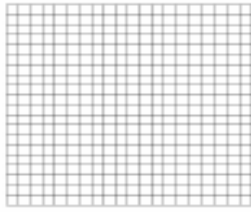
**图像数据（统一的矩形网格或体）：**一副图像是点和单元的集合体，它们分布在一个规则、矩形的格子上。格子的行、列和平面相当于世界坐标系中的  $x-y-z$  坐标。

**矩形网格（不一致的矩形网格）：**矩形数据结构是分布在一个规则的格子上的点和单元的集合。格子的行、列、平面相当于世界坐标系中的  $x-y-z$  坐标。尽管数据结构的拓扑结构规则，但其几何结构仅是部分规则。也即是说，点是按坐标进行分布，但点与点之间的距离则不尽相同。

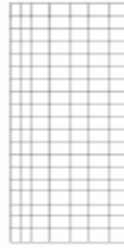
**结构化的网格（曲线网格）：**一个结构化的网格带有规则的拓扑结构和不规则的几何结构。网格可以歪曲为任何外形，其中单元不迭加或交互。结构化网格的拓扑结构通过设定一个三维的向量（ $n_x, n_y, n_z$ ）来隐式代表。几何结构通过使用一系列的点坐标来显式表示。

**非结构化网格：**大部分数据结构的格式都是非结构化网格。其拓扑和几何结构都是完全非结构化。在一个非结构化的网格里，任一个单元类型都可以以任意的组合形成。

**多边形数据：**多边形数据是几何初始量的集合体，用于绘制工具来产生显示区域显示的图像。多边形数据结构包括最高点多边形最高点，线，多边形线，多边形和三角带。



**Image Data**



**Rectilinear Grid**



**Structured Grid**



**Unstructured Grid**



**Polygonal Data**

---

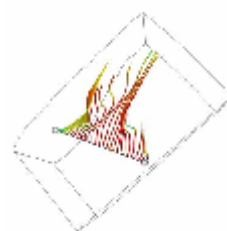
## 附录 C—三维部件

ParaView 所使用的三维部件有：

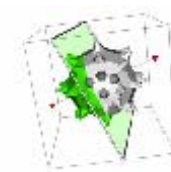
**点部件：**用于设置一个点的坐标或一个区域的中心。点的坐标可以通过点击和拖动三个坐标值的任何一个。为了更为精确的定位这个部件，你也许需要改变相机的位置。交互式按住 SHIFT 键不放，将会控制点的动作到 x、y 或 z 平面。



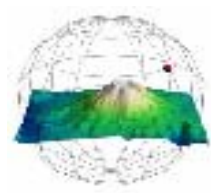
**线部件：**用于设置线的方位和位置。线的位置可以通过点击线的除了终点外的任何一个端点进行拖动进行改变。为了更为精确的定位该部件，你也需要移动相机位置。交互时按住 SHIFT 键将会严格控制 x、y 或 z 平面的动作。要移动终点之一，只需要在每一终点使用一个点部件即可，当点击时它们变为红色，并以球标记。



**面部件：**用于操纵剪切和切割平面。平面可以通过其上任一平面除了线中心之外的任一点进行拖动而移动，并行于其法线。平面法线可以通过操纵法线向量的每一终点的任一点部件而改变。



**球部件：**用于剪切和切割。球可以通过点击其上的任一点拖动而移动。球的半径通过由右键点击球拖动而改变。



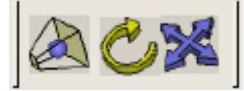
---

## 附录 D-相机控件

ParaView 提供了一些工具以使设置相机位置比较容易。它们是：

### 相机控件

首先的三个工具栏中的按钮，如右边所示，是相机控件。



第一项是能用于重置观察的按钮第二个按钮能用于在二维和三维模式之间进行切换。下面进行更为详细的描述。



重置按钮能用于重置相机以便使所有可见的项目都在观察范围之内。当前观察的方向会保持，而相机和焦点将会被调动，缩放因素将会被调整以便所有的可视对象都在观察的范围之内。项目也许仍然被另外一个对象所隐藏。



可以选中工具栏的第二个按钮来选择三维运动交互模式。通常，二维和三维交互之间的区别是，三维交互允许旋转而二维交互允许转换（panning）。一个旋转箭头用于标明三维交互模式。三维交互模式的细节可以在三维察看属性章节的相机控件中找到。



通过选择工具栏中的第三个按钮可以选中二维交互模式。一个 XY 调整箭头用于表示二维的交互模式。关于二维交互模式的细节可以在三维查看属性章节的相机章节中找到。

### 旋转控件的中心

工具栏中的最后四个按钮能够用于控制旋转中心。工具栏这个区域的显示要依靠于按钮的状态。一个可能的外形如下所示：



使用这些按钮，你

可以拾取旋转中心，重置旋转

中心，显示或隐藏标记，并且人为的编辑旋转中心的坐标值。这些操作的解释如下所知：



这个按钮能够用于交互式地拾取旋转中心。在按下次按钮后，下一个数表单击将会在显示区域拾取新的旋转中心。旋转中心的位置将会基于在鼠标位置所发现的数据。



这个按钮能够用于重置旋转中心为当前活动对象的中心。当你想围绕一个特定的数据对象旋转意欲从各种方向观察它时，这是一个非常有用的操作。



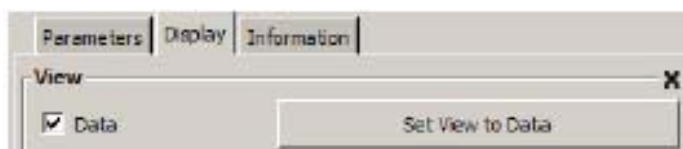
这个按钮基于模式有两种显示。当按钮显示为左方所示（中心有一个点）时，旋转中心当前可视，它们是穿过旋转中心位置中心的红色、绿色和黄色线条。按下该按钮将会引发两件事。首先，旋转中心的标志将会不可见。其次，这个按钮将会改变外观为中心不带点。按下按钮将会进行同样的反复。



这个按钮能够用于人工的编辑旋转中心。在默认情况下，它将会在工具栏中显示为斜上方带有一个小的向右的箭头。按下这个按钮，在右边当前旋转中心坐标看到的接口显示出来，并且可以使用文本进入盒进行编辑。此时，这个按钮有一个朝左的箭头，并且当按下时将会再一次隐藏文本入口盒。

## 设置数据观察

最后，你可以使用“Set View to Data”按钮来重置相机，以便活动对象完全在观察范围之内。这就相当于重置按钮，但是它仅适用于当前活动对



象。