



TVTK的管线

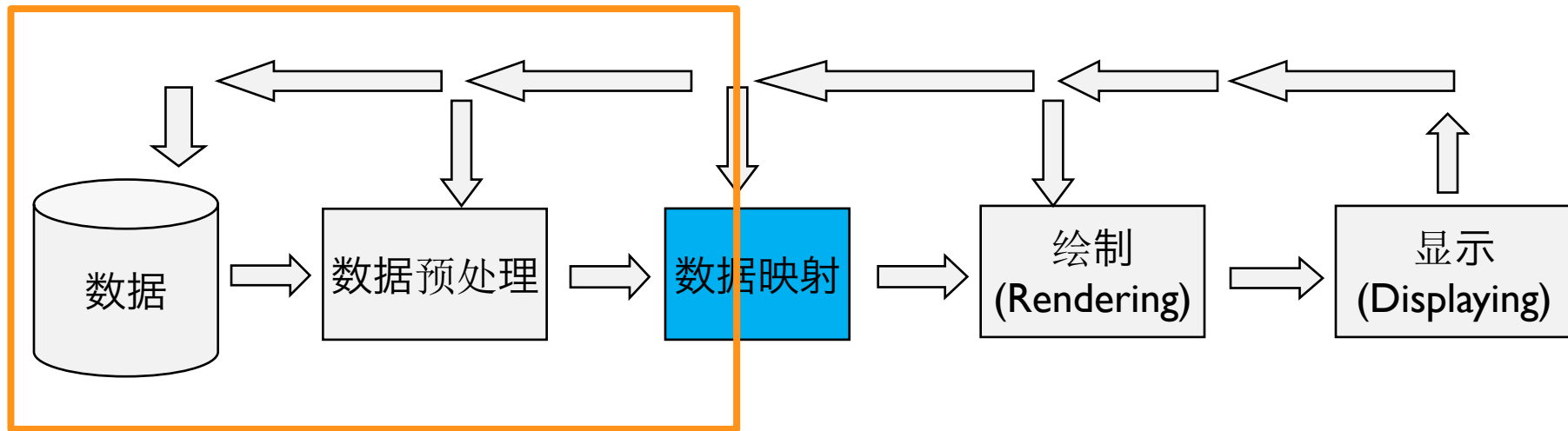
理解TVTK的管线

管线技术 (Pipeline , 流水线)

可视化管线(Visualization Pipeline) : 将原始数据加工成图形数据的过程。

图形管线(Graphics Pipeline) : 图形数据加工为我们所看到的图像的过程。

理解TVTK的管线



可视化管线

可视化管线

TVTK对象	说明
CubeSource	通过程序内部计算输出一组描述长方体的数据(PolyData)
PolyDataMapper	PolyData通过该映射器将数据映射为图形数据 (mapper)

可视化管线

```
from tvtk.api import tvtk
```

```
# 创建一个长方体数据源，并且同时设置其长宽高
```

```
s = tvtk.CubeSource(x_length=1.0, y_length=2.0, z_length=3.0)
```

```
# 使用PolyDataMapper将数据转换为图形数据
```

```
m = tvtk.PolyDataMapper(input_connection=s.output_port)
```

```
a = tvtk.Actor(mapper=m)
```

```
r = tvtk.Renderer(background=(0, 0, 0))
```

```
r.add_actor(a)
```

```
w = tvtk.RenderWindow(size=(300,300))
```

```
w.add_renderer(r)
```

```
i = tvtk.RenderWindowInteractor(render_window=w)
```

```
i.initialize()
```

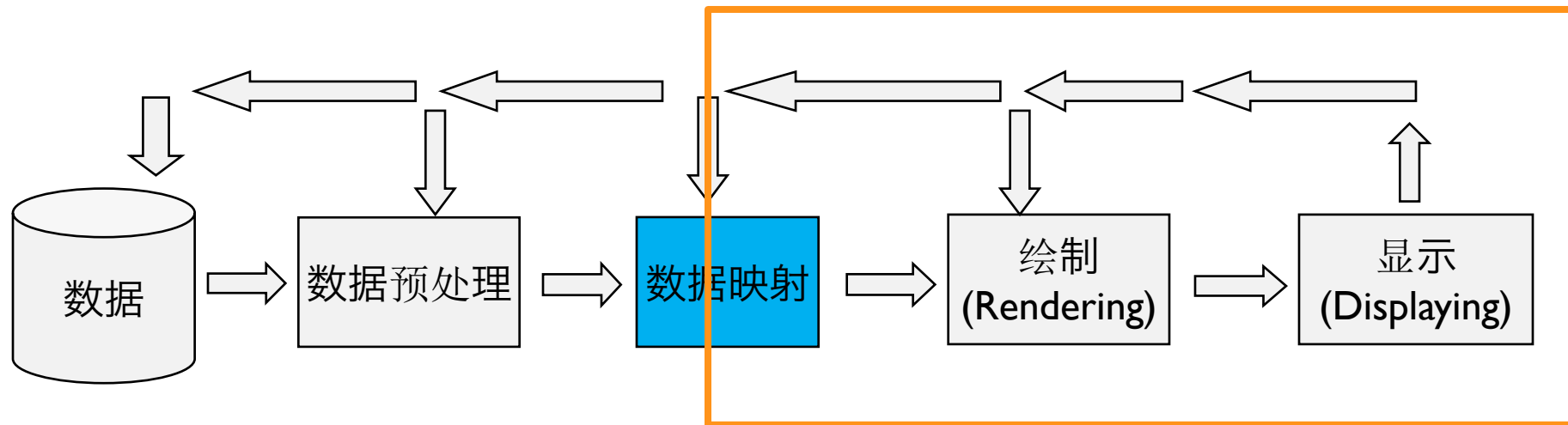
```
i.start()
```

可视化管线

```
>>> s.output_port  
<tvtk.tvtk_classes.algorithm_output.AlgorithmOutput object at 0x0000020E141F0990>  
>>> m.input_connection  
<tvtk.tvtk_classes.algorithm_output.AlgorithmOutput object at 0x0000020E141F0990>  
>>> |
```

对象经由input_connection和output_port属性连接起来

理解TVTK的管线



图形管线

图形管线

TVTK对象	说明
Actor	场景中的一个实体。它包括一个图形数据(mapper)，具有描述该实体的位置、方向、大小的属性。
Renderer	渲染的场景。它包括多个需要渲染的Actor。
RenderWindow	渲染用的图形窗口，它包括一个或者多个Render。
RenderWindowInteractor	给图形窗口提供一些用户交互功能，例如平移、旋转、放大缩小。这些交互式操作并不改变Actor或者图形数据的属性，只是调整场景中的照相机(Camera)的一些设置。

图形管线

```
from tvtk.api import tvtk
```

```
# 创建一个长方体数据源，并且同时设置其长宽高
```

```
s = tvtk.CubeSource(x_length=1.0, y_length=2.0, z_length=3.0)
```

```
# 使用PolyDataMapper将数据转换为图形数据
```

```
m = tvtk.PolyDataMapper(input_connection=s.output_port)
```

```
# 创建一个Actor
```

```
a = tvtk.Actor(mapper=m)
```

```
# 创建一个Renderer，将Actor添加进去
```

```
r = tvtk.Renderer(background=(0, 0, 0))
```

```
r.add_actor(a)
```

```
# 创建一个RenderWindow(窗口)，将Renderer添加进去
```

```
w = tvtk.RenderWindow(size=(300,300))
```

```
w.add_renderer(r)
```

```
# 创建一个RenderWindowInteractor (窗口的交互工具)
```

```
i = tvtk.RenderWindowInteractor(render_window=w)
```

```
# 开启交互
```

```
i.initialize()
```

```
i.start()
```