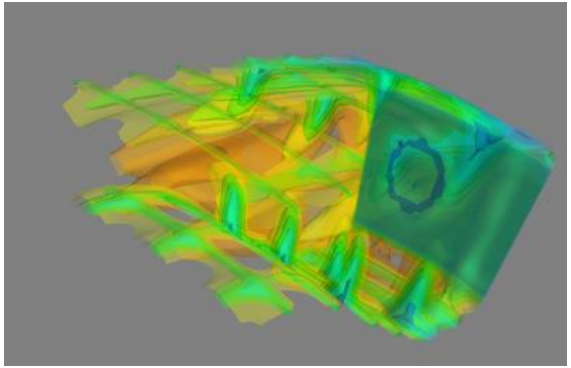
TVTK可视化实例

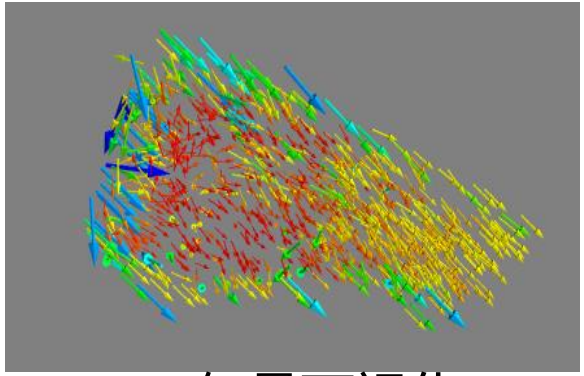
SV03

黄天羽

www.python123.org



标量可视化

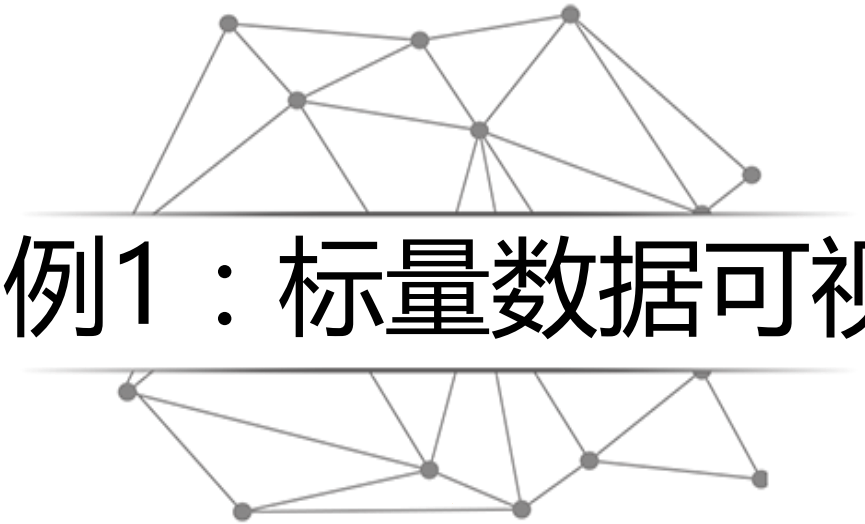


矢量可视化

轮廓线可视化

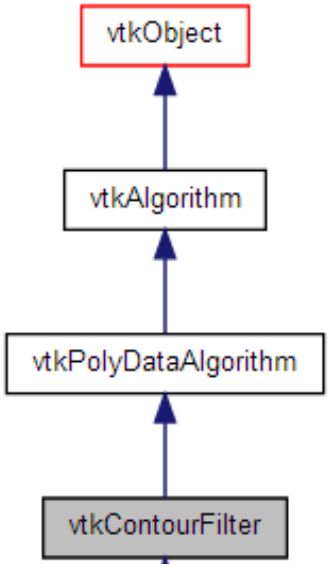
回顾Plot3D数据读取

```
from tvtk.api import tvtk
from tvtkfunc import ivtk_scene, event_loop

# 读入数据
plot3d = tvtk.MultiBlockPLOT3DReader(
        xyz_file_name="combxyz.bin",#网格文件
        q_file_name="combq.bin",#空气动力学结果文件
        scalar_function_number=100, vector_function_number=200
    )
plot3d.update()
grid = plot3d.output.get_block(0)
```



实例1：标量数据可视化

tvtk.ContourFilter()　等值面过滤器



方法	说明
generate_values()	设定n条等值线的值，一般用于重新绘制等值线
set_value()	设定一条等值线的值，一般用于覆盖某条等值线或者新增加一条等值线

数据读取

```
from tvtk.api import tvtk
from tvtkfunc import ivtk_scene

plot3d = tvtk.MultiBlockPLOT3DReader(
        xyz_file_name="combxyz.bin",
        q_file_name="combq.bin",
        scalar_function_number=100, vector_function_number=200
    )#读入Plot3D数据
plot3d.update()#让plot3D计算其输出数据
grid = plot3d.output.get_block(0)#获取读入的数据集对象
```

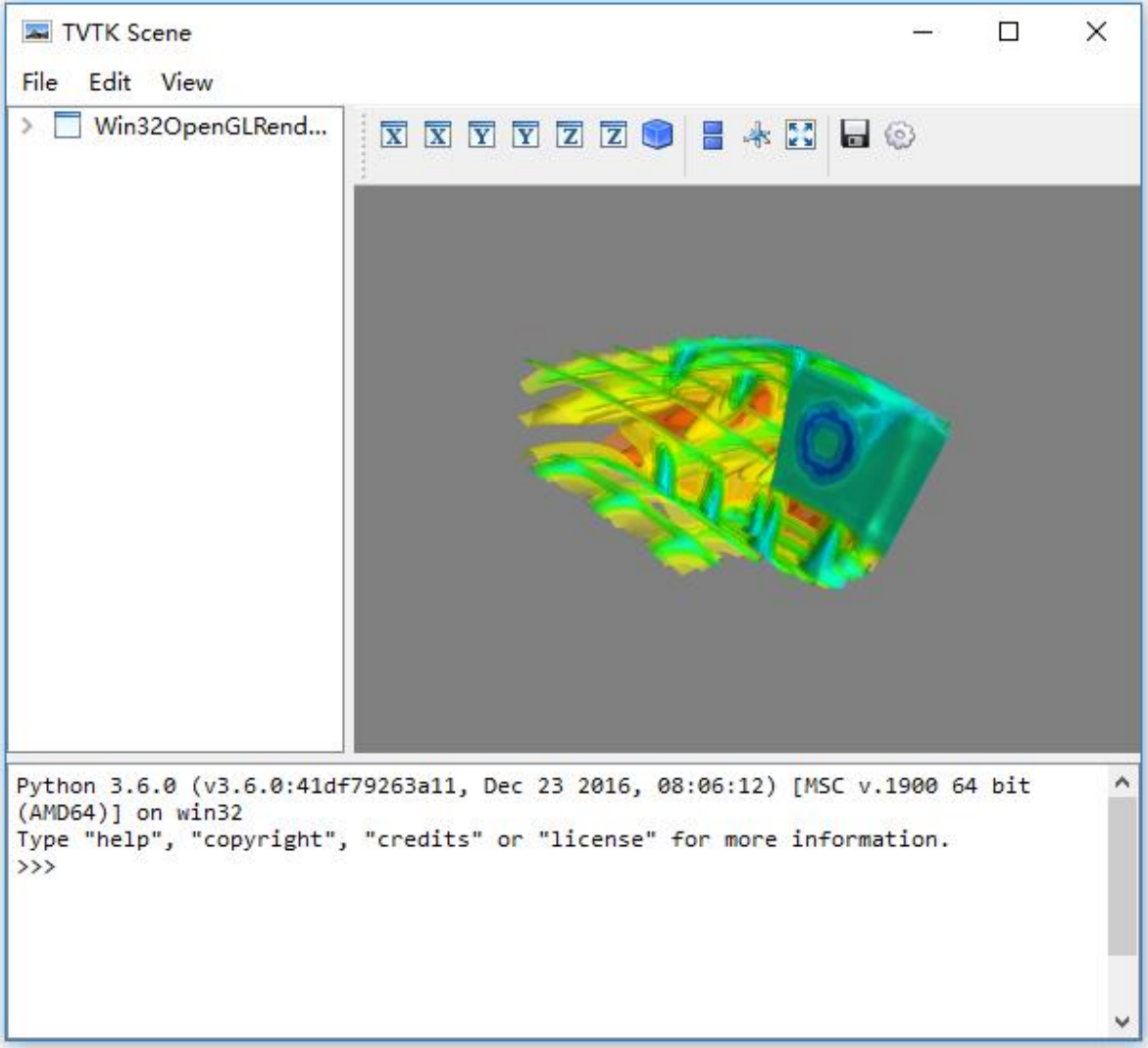
创建等值面

```
con = tvtk.ContourFilter()#创建等值面对象
con.set_input_data(grid)
con.generate_values(10, grid.point_data.scalars.range)#指定轮廓数和数据范围
#设定映射器的变量范围属性
```

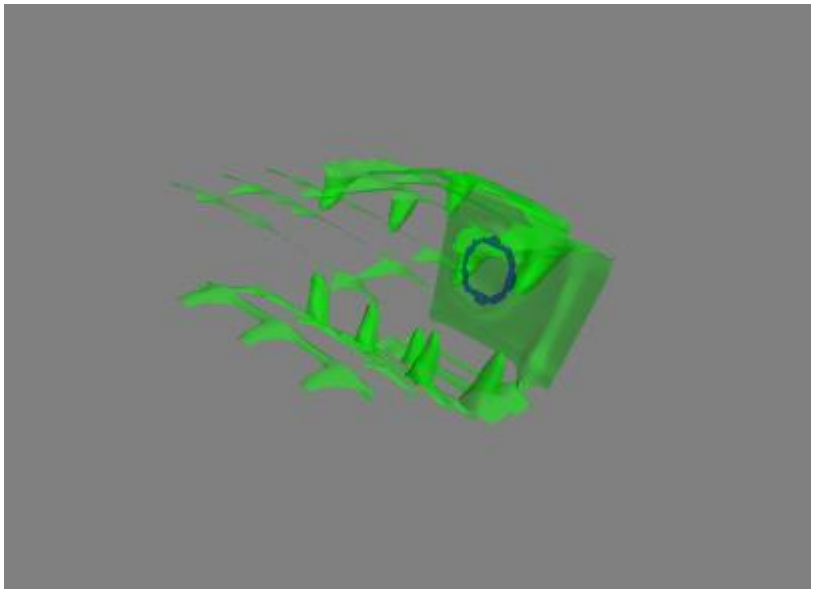
绘制数据

```
m = tvtk.PolyDataMapper(scalar_range = grid.point_data.scalars.range,
                        input_connection=con.output_port)
a = tvtk.Actor(mapper = m)
a.property.opacity = 0.5#设定透明度为0.5
#窗口绘制
win = ivtk_scene(a)
```

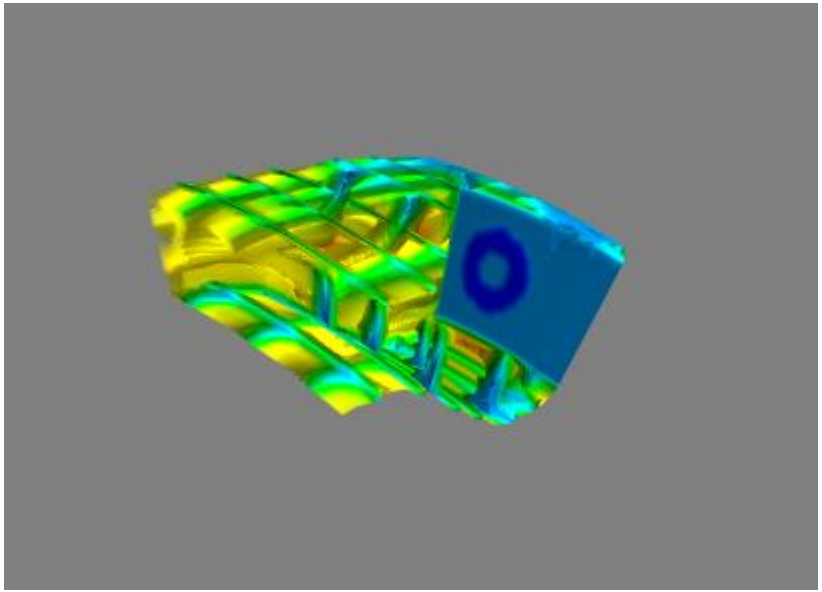
```
from tvtk.api import tvtk
from tvtkfunc import ivtk_scene, event_loop

plot3d = tvtk.MultiBlockPLOT3DReader(
        xyz_file_name="combxyz.bin",
        q_file_name="combq.bin",
        scalar_function_number=100, vector_function_number=200
    )#读入Plot3D数据
plot3d.update()#让plot3D计算其输出数据
grid = plot3d.output.get_block(0)#获取读入的数据集对象

con = tvtk.ContourFilter()#创建等值面对象
con.set_input_data(grid)
con.generate_values(10, grid.point_data.scalars.range)#指定轮廓数和数据范围
#设定映射器的变量范围属性
m = tvtk.PolyDataMapper(scalar_range = grid.point_data.scalars.range,
                        input_connection=con.output_port)
a = tvtk.Actor(mapper = m)
a.property.opacity = 0.5#设定透明度为0.5
#窗口绘制
win = ivtk_scene(a)
```


TVTK Scene
File
Edit
View
Win32OpenGLRend...
Python 3.6.0 (v3.6.0:41df79263a11, Dec 23 2016, 08:06:12) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>

```
con.generate_values(n, grid.point_data.scalars.range)
```

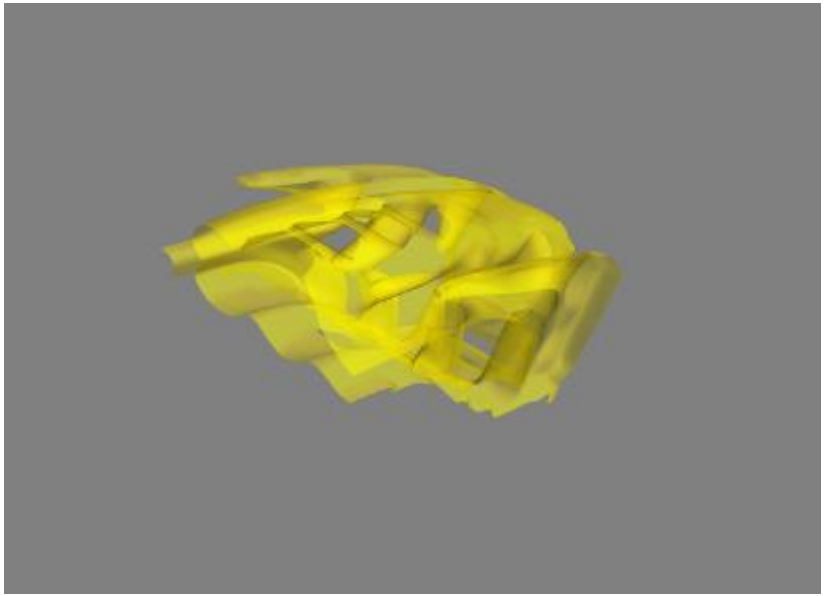


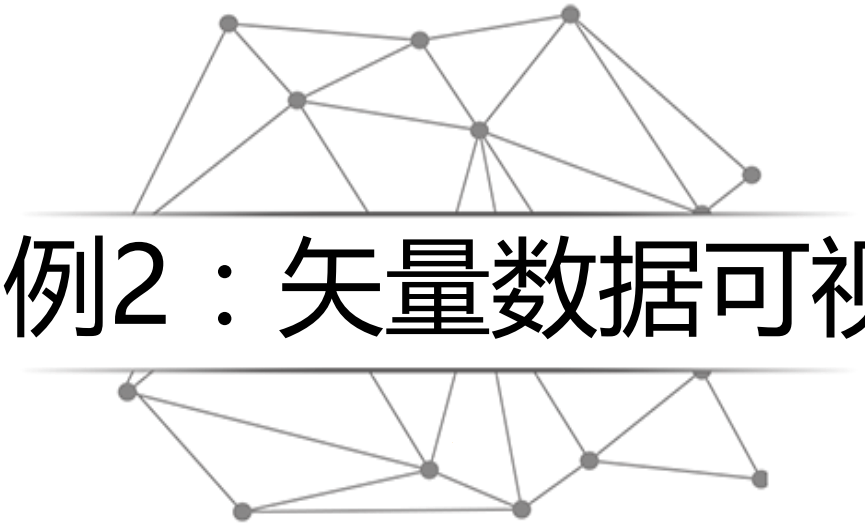
n=3（包含边界值）



n=200

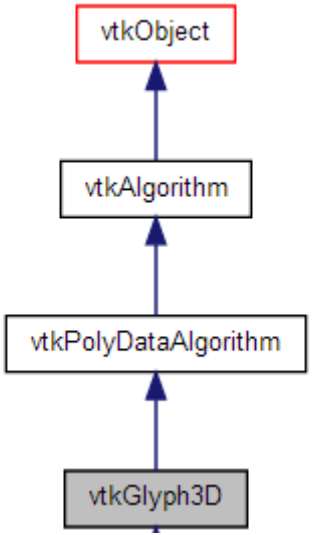
```
con.set_value(0,0.3)
```
指定等值面和对应的等值面的值



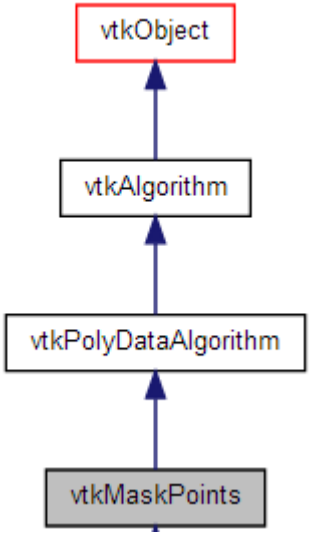


实例2：矢量数据可视化

tvtk.Glyph3D() 符号化技术



tvtk.MaskPoints () 降采样



数据读取

```
from tvtk.api import tvtk
from tvtkfunc import ivtk_scene, event_loop

#读入PLot3D数据
plot3d = tvtk.MultiBlockPLOT3DReader(
        xyz_file_name="combxyz.bin",
        q_file_name="combq.bin",
        scalar_function_number=100, vector_function_number=200
    )
plot3d.update()
grid = plot3d.output.get_block(0)
```

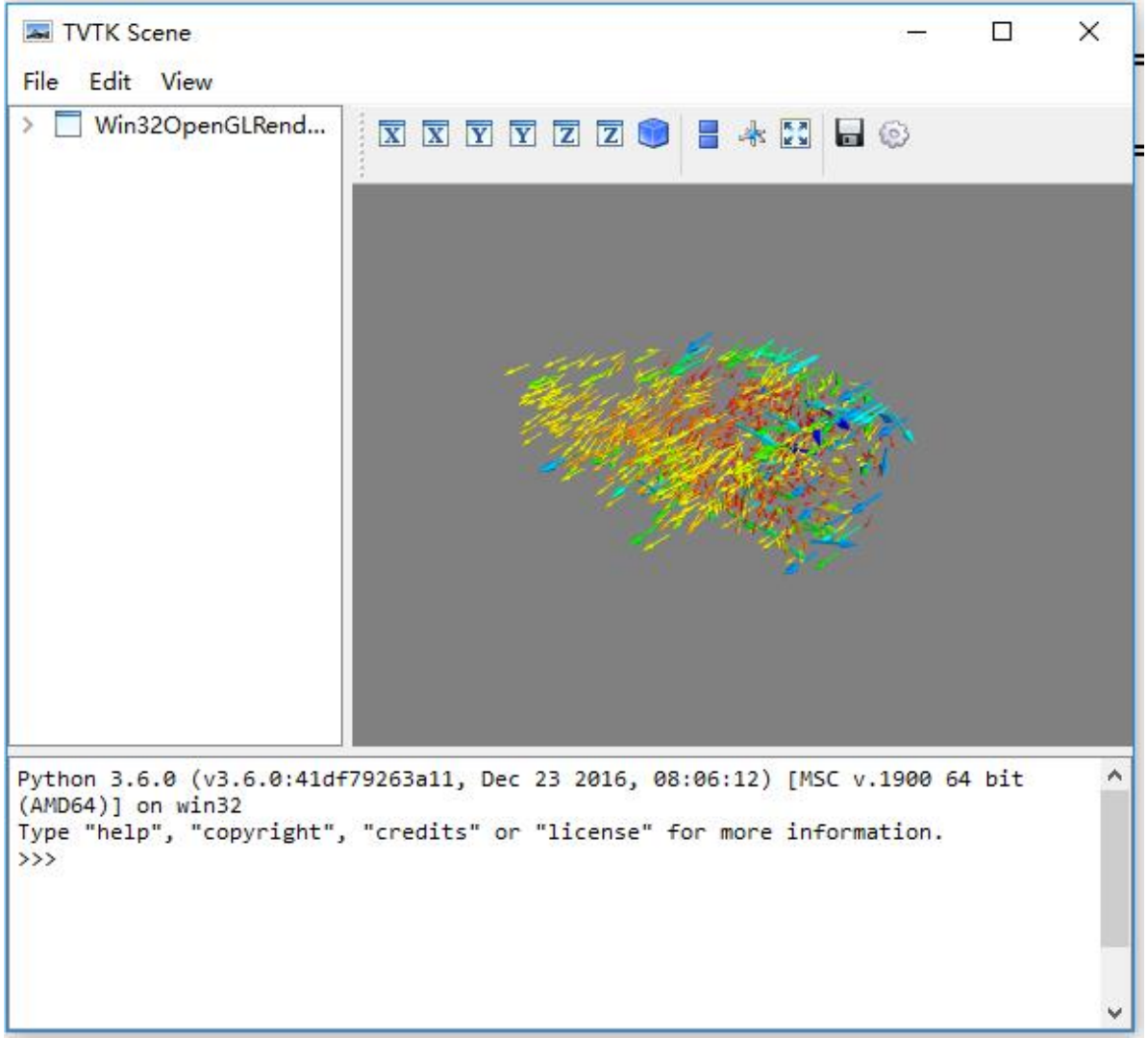
数据随机选取

```
#对数据集中的数据进行随机选取，每50个点选择一个点
mask = tvtk.MaskPoints(random_mode=True, on_ratio=50)
mask.set_input_data(grid)
#创建表示箭头的PolyData数据集
glyph_source = tvtk.ArrowSource()
```

绘制箭头

```
#在Mask采样后的PolyData数据集每个点上放置一个箭头
#箭头的方向、长度和颜色由于点对应的矢量和标量数据决定
glyph = tvtk.Glyph3D(input_connection=mask.output_port,
                     scale_factor=4)
glyph.set_source_connection(glyph_source.output_port)
m = tvtk.PolyDataMapper(scalar_range=grid.point_data.scalars.range,
                        input_connection=glyph.output_port)
a = tvtk.Actor(mapper=m)
```

```
from tvtk.api import tvtk
from tvtkfunc import ivtk_scene, event_loop

#读入PLot3D数据
plot3d = tvtk.MultiBlockPLOT3DReader(
        xyz_file_name="combxyz.bin",
        q_file_name="combq.bin",
        scalar_function_number=100, vector_function_number=200
    )
plot3d.update()
grid = plot3d.output.get_block(0)
#对数据集中的数据进行随机选取，每50个点选择一个点
mask = tvtk.MaskPoints(random_mode=True, on_ratio=50)
mask.set_input_data(grid)
#创建表示箭头的PolyData数据集
glyph_source = tvtk.ArrowSource()
#在Mask采样后的PolyData数据集每个点上放置一个箭头
#箭头的方向、长度和颜色由于点对应的矢量和标量数据决定
glyph = tvtk.Glyph3D(input_connection=mask.output_port,
                     scale_factor=4)
glyph.set_source_connection(glyph_source.output_port)
m = tvtk.PolyDataMapper(scalar_range=grid.point_data.scalars.range,
                        input_connection=glyph.output_port)
a = tvtk.Actor(mapper=m)
#窗口绘制
win = ivtk_scene(a)
win.scene.isometric_view()
event_loop()
```


TVTK Scene
File
Edit
View
Win32OpenGLRend...
Python 3.6.0 (v3.6.0:41df79263a11, Dec 23 2016, 08:06:12) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>

tvtk.Glyph3D()　符号化技术

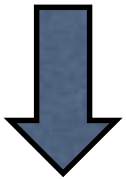
```
#对数据集中的数据进行随机选取，每50个点选择一个点
mask = tvtk.MaskPoints(random_mode=True, on_ratio=50)
mask.set_input_data(grid)
#创建表示箭头的PolyData数据集
glyph_source = tvtk.ArrowSource()
#在Mask采样后的PolyData数据集每个点上放置一个箭头
#箭头的方向、长度和颜色由于点对应的矢量和标量数据决定
glyph = tvtk.Glyph3D(input_connection=mask.output_port,
                     scale_factor=4)
glyph.set_source_connection(glyph_source.output_port)
```

tvtk.MaskPoints () 降采样

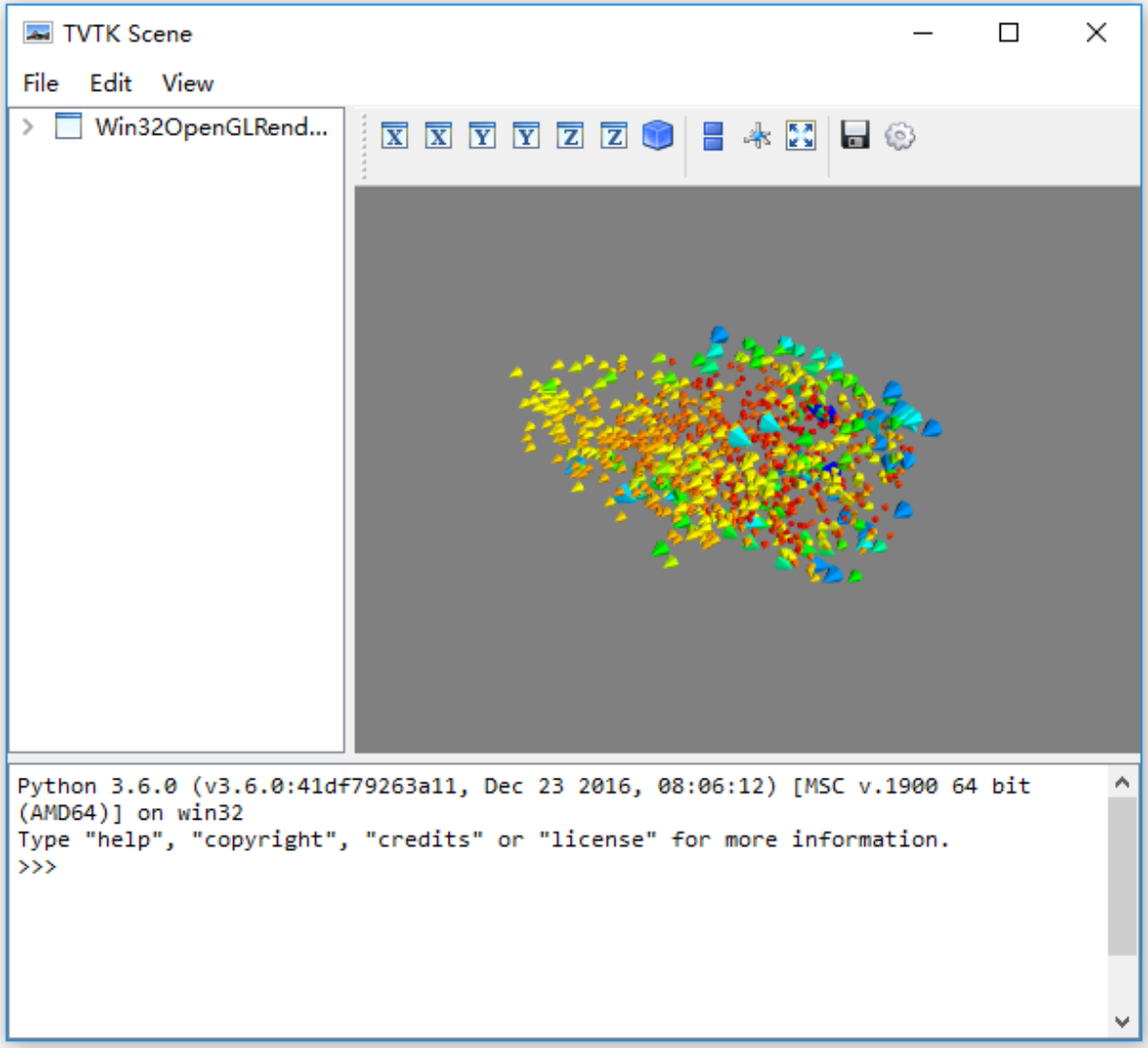
```
#对数据集中的数据进行随机选取，每50个点选择一个点
mask = tvtk.MaskPoints(random_mode=True, on_ratio=50)
mask.set_input_data(grid)
#创建表示箭头的PolyData数据集
glyph_source = tvtk.ArrowSource()
#在Mask采样后的PolyData数据集每个点上放置一个箭头
#箭头的方向、长度和颜色由于点对应的矢量和标量数据决定
glyph = tvtk.Glyph3D(input_connection=mask.output_port,
                     scale_factor=4)
glyph.set_source_connection(glyph_source.output_port)
```

```
>>> print(grid.number_of_points)
47025
>>> mask.update()
>>> print(mask.output.number_of_points)
930
>>>
```

glyph_source = tvtk.ArrowSource()



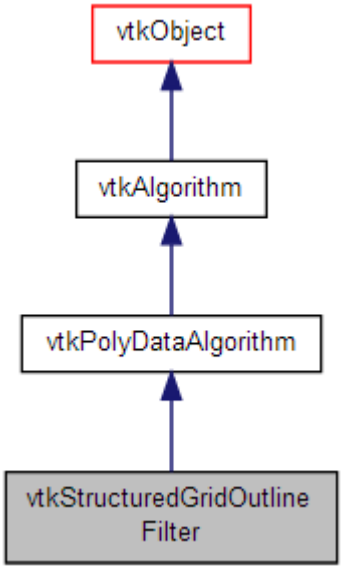
glyph_source = tvtk.ConeSource()
设置放缩系数：scale_factor = 2


TVTK Scene
File
Edit
View
Win32OpenGLRend...
Python 3.6.0 (v3.6.0:41df79263a11, Dec 23 2016, 08:06:12) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>

实例3：空间轮廓线可视化

tvtk.StructuredGridOutlineFilter()

计算PolyData对象的外边框。



数据读取

```
from tvtk.api import tvtk
from tvtk.common import configure_input
from tvtkfunc import ivtk_scene, event_loop

plot3d = tvtk.MultiBlockPLOT3DReader(
        xyz_file_name="combxyz.bin",
        q_file_name="combq.bin",
        scalar_function_number=100, vector_function_number=200
    )#读入Plot3D数据
plot3d.update()#让plot3D计算其输出数据
grid = plot3d.output.get_block(0)#获取读入的数据集对象
```

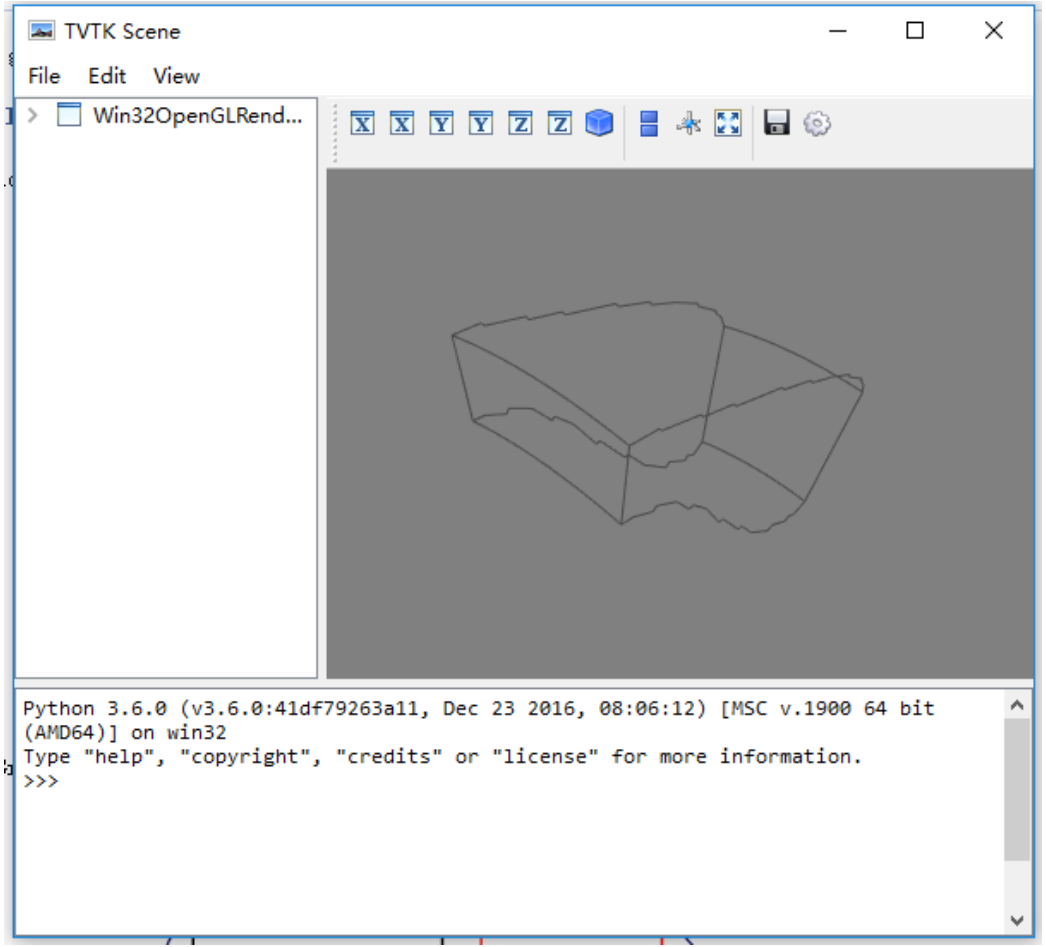
计算轮廓线

```
tvtk.StructuredGridOutlineFilter()
```

计算轮廓线

```
outline = tvtk.StructuredGridOutlineFilter()#计算表示外边框的PolyData对象
configure_input(outline, grid)#调用tvtk.common.configure_input()
m = tvtk.PolyDataMapper(input_connection=outline.output_port)
a = tvtk.Actor(mapper=m)
a.property.color = 0.3, 0.3, 0.3
```

```
from tvtk.api import tvtk
from tvtk.common import configure_input
from tvtkfunc import ivtk_scene, event_loop

plot3d = tvtk.MultiBlockPLOT3DReader(
        xyz_file_name="combxyz.bin",
        q_file_name="combq.bin",
        scalar_function_number=100, vector_function_number=200
    )#读入Plot3D数据
plot3d.update()#让plot3D计算其输出数据
grid = plot3d.output.get_block(0)#获取读入的数据集对象
outline = tvtk.StructuredGridOutlineFilter()#计算表示外边框的PolyData对象
configure_input(outline, grid)#调用tvtk.common.configure_input()
m = tvtk.PolyDataMapper(input_connection=outline.output_port)
a = tvtk.Actor(mapper=m)
a.property.color = 0.3, 0.3, 0.3

#窗口绘制
win = ivtk_scene(a)
win.scene.isometric_view()
event_loop()
```


TVTK Scene
File
Edit
View
Win32OpenGLRend...
Python 3.6.0 (v3.6.0:41df79263a11, Dec 23 2016, 08:06:12) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>