

TP Visualization Toolkit

julien.finet@kitware.com

Le but de ce TP est de se familiariser avec la bibliothèque VTK (Visualization Toolkit) et d'utiliser les méthodes de visualisation présentes dans cette bibliothèque. VTK possède une documentation disponible à <http://www.vtk.org>).

Exercice 1 : VTK Hello World

Le but de cet exercice est d'écrire un script Python qui affiche la version d'VTK installée. Cette application permet juste de vérifier que VTK est bien installé et fonctionne correctement.

Astuces :

1. On importera le module python **vtk**
2. Il faudra chercher dans la documentation (ou sur le web) comment afficher la version du package.

Exercice 2 : Charger et Afficher un modèle 3D

Ecrire un script python qui permet de lire et d'afficher le polydata cow.vtk

Astuces :

1. Il faudra utiliser un PolyDataReader ainsi qu'un PolyDataMapper
2. Les composants suivants seront nécessaires : vtkActor, vtkRenderer, vtkRenderWindow, vtkRenderWindowInteractor

Questions :

1. Comment peut-on afficher la donnée de couleur rose (255,192,203) en RGB?

Exercice 3 : Filtrage de données

A partir du script précédent, nous souhaitons avoir un rendu plus « lisse ».

Astuces :

1. La première étape est purement du rendu en ajoutant des normals au PolyData
2. La deuxième étape est d'utiliser un filtre de Smoothing du PolyData

Questions :

1. Essayez de réduire la donnée avec vtkDecimatePro
2. Combien avons-nous de sources dans le pipeline ?
3. Combien avons-nous de filtres ?

4. Combien avons-nous de mapper ?
5. Combien avons-nous de « Data Objects » ?

Exercice 4 : Création d'un PolyData

Développer un script qui génère et affiche un triangle (polyData)

Astuces :

1. Il faudra créer trois points et ensuite une cellule avec ces trois points
2. Utilisez le SetPolys() fonction de vtkPolyData

Exercice 5 : Visualisation en Rendu Volumique

Développer un script qui charge et affiche en rendu volume la donnée « poship.slc »

Astuces :

1. On utilisera la classe vtkSLCReader
2. On utilisera un vtkSmartVolumeMapper ainsi qu'un vtkVolume
3. Afin d'avoir un meilleur rendu nous utiliserons les fonctions de transfert suivantes :

```
opacity_transfer_function = vtk.vtkPiecewiseFunction()
opacity_transfer_function.AddPoint(20, 0.0)
opacity_transfer_function.AddPoint(255, 0.3)

color_transfer_function = vtk.vtkColorTransferFunction()
color_transfer_function.AddRGBPoint(0.0, 0.0, 0.0, 0.0)
color_transfer_function.AddRGBPoint(64.0, 1.0, 0.0, 0.0)
color_transfer_function.AddRGBPoint(128.0, 0.0, 0.0, 1.0)
color_transfer_function.AddRGBPoint(192.0, 0.0, 1.0, 0.0)
color_transfer_function.AddRGBPoint(255.0, 0.0, 0.2, 0.0)

volume_property = vtk.vtkVolumeProperty()
volume_property.SetColor(color_transfer_function)
volume_property.SetScalarOpacity(opacity_transfer_function)
```

Exercice 6 : Visualisation de contours d'une image 3D

Développer un script qui charge et affiche **le contour à la valeur scalaire 135** la donnée « head.vti »

Exercice 7 : Visualisation volumique d'une image 3D

A partir de l'exercice 6, ajouter le rendu volumique de l'image 3D en plus du contour.

Astuces :

1. On pourra utiliser la fonction de transfert d'opacité avec les valeurs suivantes :

```
opacityFun.AddPoint(0.0, 0.0)
opacityFun.AddPoint(90.0, 0.0)
opacityFun.AddPoint(100.0, 0.2)
opacityFun.AddPoint(120.0, 0.0)
```

et les couleurs suivantes :

```
colorFun.AddRGBPoint(0.0, .8, .4, .2)
colorFun.AddRGBPoint(255.0, .8, .4, .2)
```

Exercice 8 : Connection d'ITK et VTK

Ecrire un script qui charge la donnée brain.mhd avec ITK et qui l'affiche avec VTK

Astuces :

1. On utilisera itk.imread() et itk.vtk_image_from_image()
2. Soyez créatif sur la visualisation 😊