

VTK Projet - alexandre.nicaise28@gmail.com - Gmail

VTK Projet

Participants: alexandre.nicaise28@gmail.com, jean-luc.mari@univ-amu.fr, florianbeguet@orange.fr, s.sarahbali@gmail.com, arnaud.polette@univ-amu.fr

Ouvrir la conversation dans Gmail

Florian BEGUET

3 juillet 2015 18:02

Bonsoir,

Je suis désolé d'avoir mis autant de temps à te répondre :s

Alors le projet consistait à élaborer une méthode 3D de détection automatique des cratères sur un maillage d'astéroïde et nous avons utilisé le langage C++.

Pour la visualisation c'est exactement ce qu'il faut faire pour afficher directement un rendu (j'imagine que tu as utilisé aussi un "vtkActor" dans le "vtkRenderWindow").

Tu as la possibilité d'utiliser différentes palettes de couleur prédéfinies avec "vtkColorSeries" et en utilisant SetColorScheme() pour déterminer ta palette préférée.

```
/*scheme détermine la palette de couleur à utiliser(voir doc vtk pour le choix)*/
```

```
int scheme = 16;
```

```
vtkSmartPointer<vtkColorSeries> colorSeries = vtkSmartPointer<vtkColorSeries>::New();  
colorSeries->SetColorScheme(scheme);
```

```
vtkSmartPointer<vtkColorTransferFunction> lut = vtkSmartPointer<vtkColorTransferFunction>::New();  
lut->SetColorSpaceToHSV();
```

```
int numColors = colorSeries->GetNumberOfColors();
```

```
for (int i = 0; i < numColors; i++)
```

```
{
```

```
    vtkColor3ub color = colorSeries->GetColor(i);
```

```
    double dColor[3];
```

```
    dColor[0] = static_cast<double> (color[0]) / 255.0;
```

```
    dColor[1] = static_cast<double> (color[1]) / 255.0;
```

```

    dColor[2] = static_cast<double> (color[2]) / 255.0;
    double t = scalarRange[0] + (scalarRange[1] - scalarRange[0]) / (numColors - 1) * i;
    lut->AddRGBPoint(t, dColor[0], dColor[1], dColor[2]);
}

vtkSmartPointer<vtkPolyDataMapper> mapper = vtkSmartPointer<vtkPolyDataMapper>::New();
mapper->SetInputConnection(curvaturesFilter->GetOutputPort());
mapper->SetLookupTable(lut);
mapper->SetScalarRange(scalarRange);

/*Affichage de la barre de couleur*/
vtkSmartPointer<vtkScalarBarActor> scalarBar = vtkSmartPointer<vtkScalarBarActor>::New();
scalarBar->SetLookupTable(mapper->GetLookupTable());
scalarBar->SetTitle(curvaturesFilter->GetOutput()->GetPointData()->GetScalars()->GetName());
scalarBar->SetNumberOfLabels(5);

```

Tu peux grâce à un "vtkRenderWindowInteractor" interagir directement avec tes données (zoom, rotation, ect ...) et tu peux ajouter un "InteractorStyle" personnalisable qui te permettra de fabriquer soit du picking ou des interactions avec le clavier (la touche + permettrait par exemple d'augmenter la valeur d'un point que tu as sélectionné).

Le lien direct vers un exemple d'utilisation en C++ pour du picking.

<http://www.vtk.org/Wiki/VTK/Examples/Cxx/Picking/CellPicking>

Et voici le lien de la doc et celui où tu peux retrouver plein d'exemples en C++ comme celui pour InteractorStyle.

<http://www.vtk.org/Wiki/VTK/Examples/Cxx>

<http://www.vtk.org/doc/nightly/html/index.html>

Je te souhaite bon courage pour la suite et n'hésite pas à nous envoyer un mail dès que tu coince sur quelque chose

```

> Message du 30/06/15 17:08
> De : "Alexandre Nicaise" <alexandre.nicaise28@gmail.com>
> A : "Florian BEGUET" <florianbeguet@orange.fr>, "Sarah Bali" <s.sarahbali@gmail.com>
> Copie à : "Jean-Luc Mari" <jean-luc.mari@univ-amu.fr>, "Arnaud Polette" <arnaud.polette@univ-amu.fr>
> [ VTKPresentation.pdf (54.1 Ko) ]

```