

内部でOpenGLを使う 3D 描画エンジンVTK

橋口 大崇

古くから使われている3D描画エンジンであるVTK (The Visualization Toolkit) を使った実装について紹介します。

特徴

VTKはアカデミック分野で利用されるアプリケーションの基盤として現在も活用されています。QtなどのGUI開発プラットフォームとの連携機能を備えているため、利用頻度の高いソフトウェア・ライブラリです。

VTKは内部的にOpenGLというライブラリを用いてレンダリング処理を行っています。OpenGLは単体でも描画エンジンとして使用できますが、PCのグラフィックス・ハードウェアを直接制御したり、詳細なリソース管理が必要となったり、複雑な実装が必要となるため、習得難易度が高いという問題があります。VTKを使用することで、OpenGLの複雑な部分を抽

象化し、より簡単に3D描画を行うことが可能になります。つまり、VTKはOpenGLをより手軽に利用するためのツールとして機能します。

実装1…VTKによるPLY点群描画

本章で紹介するプログラムは、本誌Webページから提供します(VTK.ipynb)。Google Colaboratory (以下、Colab) にアップロードするだけで体験できます。

<https://www.cqpub.co.jp/interface/download/contents.htm>

加えてPLYフォーマットのファイルをアップロードしておきます。VTK.ipynb中の下記の部分を、読み込みたいファイル名に変更してください。

```
# PLYファイルを読み込む
ply_path = "/(パス名)/Cloud.ply"
```

リスト1にColab上で実行可能なVTKによるPLY点群描画の実装例を示します。リスト1のコードを実

リスト1 VTKを使ってply点群データを描画するPythonコード

```
!pip install vtk
!apt-get install -qq xvfb libgl1-mesa-glx
!pip install pyvista -qq

import vtk
from IPython.display import Image, display
import pyvista

# pyvistaのJupyterバックエンド設定
pyvista.set_jupyter_backend('static')
pyvista.global_theme.notebook = True
pyvista.start_xvfb()

# PLYファイルを読み込む
ply_path = "/content/drive/My Drive/Cloud.ply"
ply_reader = vtk.vtkPLYReader()
ply_reader.SetFileName(ply_path)
ply_reader.Update()

# 頂点情報を取り出す
points = vtk.vtkVertexGlyphFilter()
points.SetInputData(ply_reader.GetOutput())
points.Update()

# マップを作成
mapper = vtk.vtkPolyDataMapper()
mapper.SetInputConnection(points.GetOutputPort())
```

読み込みたい
ファイル名に
変更する

```
# アクタを作成
actor = vtk.vtkActor()
actor.SetMapper(mapper)
# 頂点サイズ設定
actor.GetProperty().SetPointSize(3)

# レンダラを作成
renderer = vtk.vtkRenderer()
renderer.AddActor(actor)
# 背景色を設定
renderer.SetBackground(0.9, 0.9, 0.9)

# レンダリング・ウィンドウを作成
render_window = vtk.vtkRenderWindow()
# オフスクリーン・レンダリングを有効にする
render_window.SetOffScreenRendering(1)
render_window.AddRenderer(renderer)
# ウィンドウ・サイズの設定
render_window.SetSize(600, 800)

# レンダリングとスクリーンショットの取得
render_window.Render()

# スクリーンショットを取得して画像として保存
filter = vtk.vtkWindowToImageFilter()
```