VTK-Unity

Medical Volume Viewer

v2.0.0 - Google Docs Link

Contact: kitware.eu/contact/

Bug tracker: VTKUnity-BugTracker

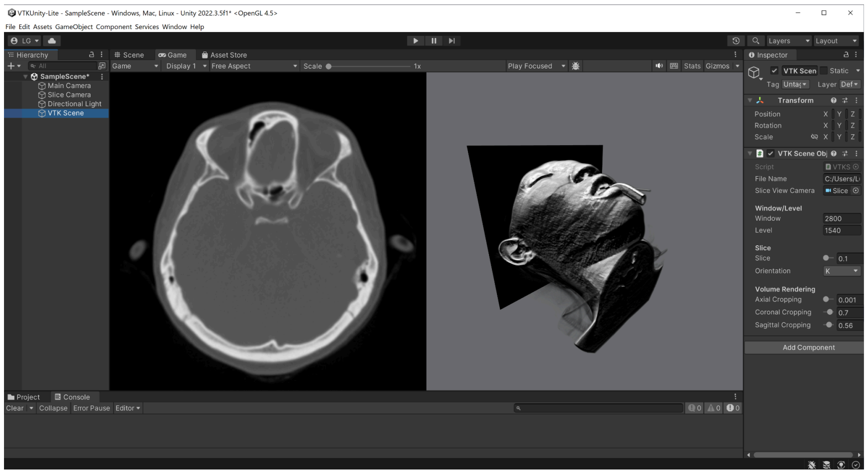
Table of contents

Overview

The role of this free asset is to demonstrate the Visualization Toolkit (VTK) rendering and processing capabilities inside a Unity application. It provides the minimal infrastructure required to create a medical image viewer in Unity. The VTK scene is rendered into Unity's rendering pipeline using Unity low-level native plug-in interface and leveraging the use of command buffers.

The asset is organized with the following directory structure:

- **Plugins**: Native and managed code exposing a subset of the VTK C++ API in Unity.
- **Scripts**: MonoBehaviour scripts used to synchronize and update VTK rendering.
- **StreamingAssets**: Data files necessary for the demo scene. You might need to **copy** the contents of this inner directory to the **StreamingAssets** directory located in the **Assets** folder of your project.
- **Scenes**: Example scene using input files from the StreamingAssets directory to showcase VTK visualization and computation capabilities. An exhaustive description of the features available in the demo scene is given in section 4.



VTK Scene | Static | Tag Untagged | Layer Default

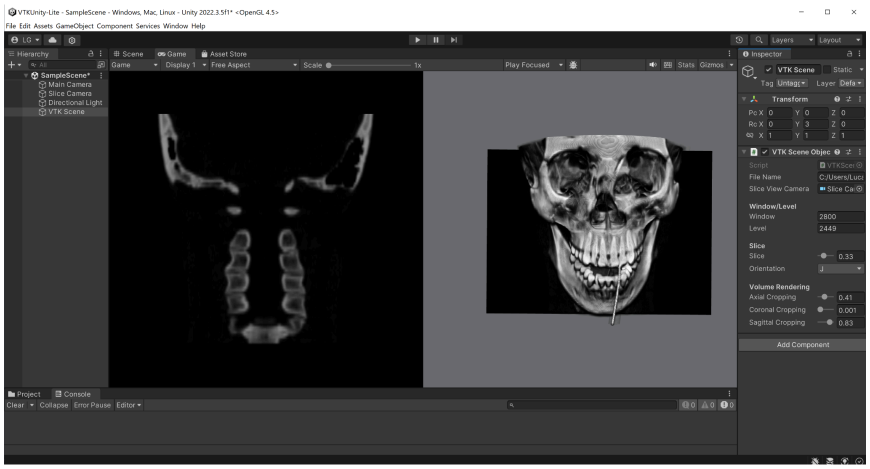
Transform: Position X Y Z; Rotation X Y Z; Scale X Y Z

VTK Scene Obj
Script: VTKS
File Name: C:/Users/L
Slice View Camera: Slice

Window/Level
Window: 2800
Level: 1540

Slice
Slice: 0.1
Orientation: K

Volume Rendering
Axial Cropping: 0.001
Coronal Cropping: 0.7
Sagittal Cropping: 0.56



VTK Scene | Static | Tag Untagg | Layer Defa

Transform: Po X 0 Y 0 Z 0; Ro X 0 Y 3 Z 0; X 1 Y 1 Z 1

VTK Scene Objec
Script: VTKScen
File Name: C:/Users/Luca
Slice View Camera: Slice Ca

Window/Level
Window: 2800
Level: 2449

Slice
Slice: 0.33
Orientation: J

Volume Rendering
Axial Cropping: 0.41
Coronal Cropping: 0.001
Sagittal Cropping: 0.83

Limitations

- **OS support**: The native plugin has been implemented and tested for Windows only. Support for Linux, MacOS, and Android might come in future versions.

- **Unity Graphics API**: VTK relies on the OpenGL rendering backend, thus Unity graphics API must be set to OpenGLCore.

- **Missing features**: This asset only provides a small subset of the VTK features available in C++. While future versions of this asset plan to extend the list of available features, it is possible to access the whole set of features offered by VTK by using Activiz which provides VTK API in C#. Please contact kitware@kitware.fr for details.
https://assetstore.unity.com/packages/essentials/tutorial-projects/vtkunity-activiz-163686

- **Virtual and Augmented Reality**: Although VTK and ActiViz natively support rendering in VR and AR headsets such as the Hololens 2, the Unity support for OpenXR does not work with the OpenGLCore rendering backend. For this reason, the plugin only supports rendering in VR headsets using the Unity legacy VR support that is based on OpenVR and available in **Unity 2019.**

Contact

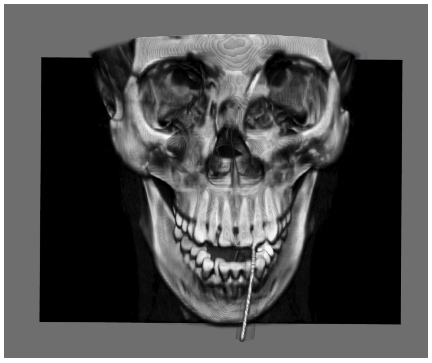
Let us know how to improve Unity support so it better fits your needs at kitware@kitware.fr, or use the bug tracker to report issues.

Example Scene description

The following features are presented in the example scene shipped with this asset and can be accessed through the VTKSceneObject.cs script. The public variables of the script are exposed in the Unity Editor to control the different features offered by the example scene. An exhaustive description is given below.

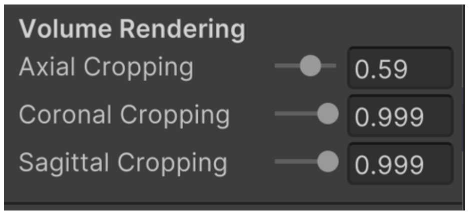
Volume Rendering

Volume Rendering is used to display a series of 2D slices as a 3D volume.



- **Volume File Name**: Specify the filename of the 3D input image (volume) to visualize. The filename can be relative to the StreamingAssets folder located in the Assets folder of the Unity project. The following extensions are supported as input:
    - VTK XML Image (.vti)
    - MetaImage (.mha/.mhd)
    - DICOM folder (Path to a folder containing DICOM images)
    - NifTI (.nii, .nii.gz)
    - Nrrd (.nrrd, .nhdr)

- **Volume Cropping:** Move the cropping sliders to control the amount of cropping applied to the volume in each anatomical direction. A value of 0.0 will disable cropping in every direction, while a value of 1.0 results in the volume being fully cropped for this specific direction.



Volume Rendering

Axial Cropping	0.59
Coronal Cropping	0.999
Sagittal Cropping	0.999

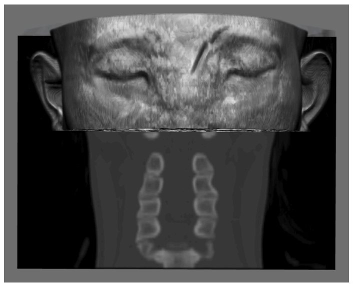
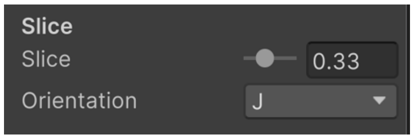


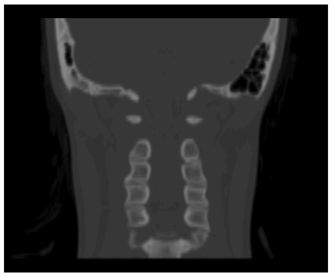
Image Slice

It is possible to display the anatomical planes of the input volume.

The slice index can be changed interactively by moving the “Slice” slider. A value of 0.0 will compute and render the first slice in the current anatomical direction, while a value of 1.0 will present the last slice.

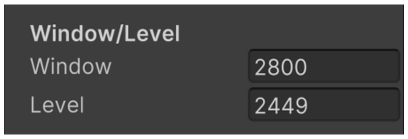
The current anatomical direction of the slice can be changed using the dropdown button of the “Orientation” parameter.

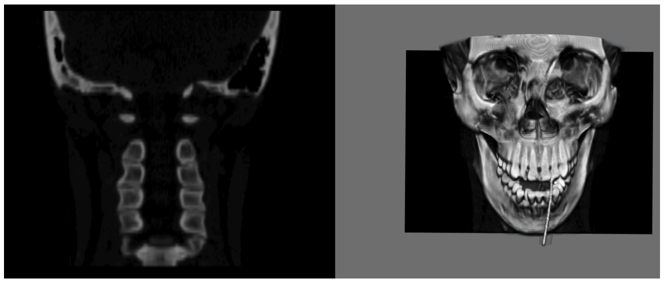

Slice
Slice 0.33
Orientation J

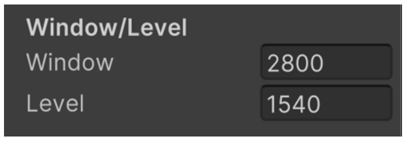


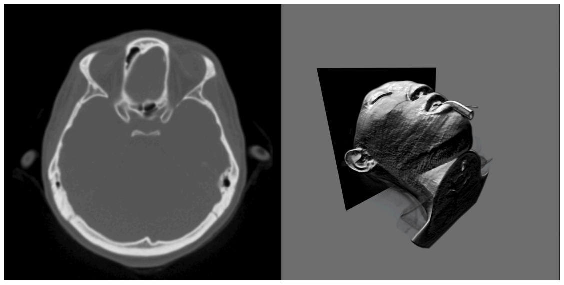
Window/Level

Window/Level, also known as windowing or contrast modification, defines the mapping of the input images grayscale values to black and white colors.
The brightness of the image is adjusted using the window level, while the contrast is adjusted using the window width.


Window/Level
Window 2800
Level 2449




Window/Level
Window 2800
Level 1540



How to use

To use the plugin, create a new Unity 3D project and proceed as follow:

- Change rendering backend to OpenGLCore
  - Edit->Project Settings->Player->Other Settings
  - Untick "Auto Graphics API for Windows"
  - Remove current API and add "OpenGLCore"

To try the sample scene:

- Import the VTKUnity-MedicalVolumeViewer package and open the sample scene in the Scene folder.

To start a project from scratch:

- Copy the Plugin folder into the Assets folder of your project
- Copy the Scripts folder into the Assets folder of your project
  - Add VTKCamera.cs to the main Camera of your Unity project
  - Add VTKLight.cs to every light component in your Unity project
  - Add VTKSceneObject.cs to an empty game object in the Unity scene
- Copy input images to be visualized in the StreamingAssets folder located in the Assets folder of the unity project.

Monobehaviour scripts components

The following script components are provided for users to set up a VTK scene within their Unity project.

Scripts within the Core folder are provided to synchronize the two scenes:

- *VTKCamera.cs*: This script should be attached to Unity's Camera to trigger VTK rendering. It uses Unity's command buffers to call the VTK rendering callback implemented in the native plugin. It exposes a vtkRenderer object in which the VTK scene objects can be rendered.

- *VTKLight.cs*: This script should be added to a Unity light object to set up the corresponding VTK scene light and synchronize light parameters. VTK supports illumination with Directional, Point and Spot light types. Shadows, indirect lighting and cookies are unsupported for now.

- *VTKSceneObject.cs*: The VTKSceneObject.cs script implements a MonoBehavior subclass that provides callback functions being called for each camera having a VTKCamera component. Such callbacks allow users to easily add and remove scene objects from the associated vtkRenderer.

- *VTKSceneCamera.cs*: This script can be attached to an empty game object to trigger rendering in the editor without starting the game. This requires all scripts to use the [ExecuteInEditMode] keyword.

- *VTKNativePlugin.cs*: This script provides helper functions called internally to synchronize the VTK render passes with the Unity render pipeline.

- *VTKNativePluginLite.cs*: Additional helper functions exposing a subset of the VTK C++ API in Unity in order to manage the VTK rendering stack objects such as the camera.

- *VTKNativePluginLiteScene.cs*: This script exposes a subset of the VTK C++ API in Unity in order to add, remove, and manage the VTK scene elements. The full description of features provided in the sample scene is available in the next section.

License

VTK is an open-source toolkit distributed under the OSI-approved BSD 3-clause License. See the following Copyright.txt for details.

Unset