

3D Stereo Cursor in VTK

By Kitware Europe - V1 - July 2023 - contact: kitware@kitware.fr

3D Stereo Cursor Concept	2
Design Guide	3
Source Code and Classes Documentations	3
Overall Design	3
Usage in a VTK-based Application	5
Integration Examples	6
ParaView	6
3D Slicer	8

3D Stereo Cursor Concept

When working with stereoscopic content, the shape and behavior of a standard 2D mouse cursor is not suitable, as depth conflicts between the cursor and the content can lead to confusion.

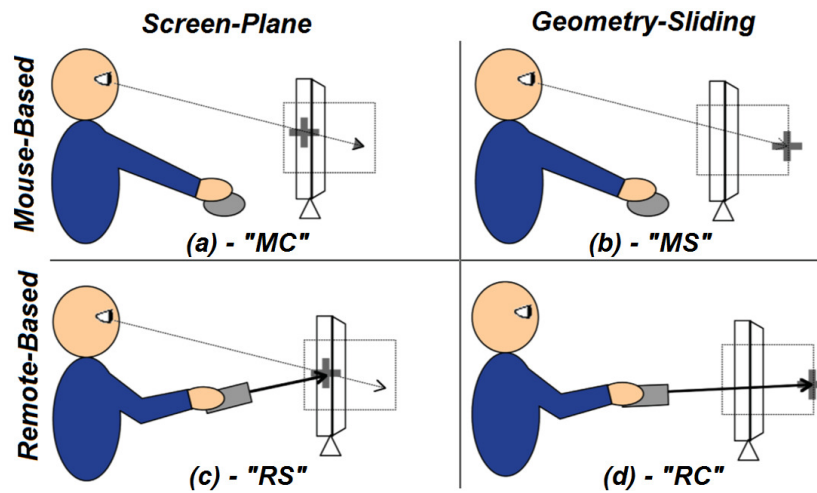


Figure 1. Four 3D pointing techniques, from Robert J. Teather, Wolfgang Stuerzlinger "Pointing at 3D Target Projections with One-Eyed and Stereo Cursors"

This document demonstrates the 3D Stereo Cursor solution in VTK, including the chosen design for the implementation and the typical usage in a VTK-based application.

Design Guide

Source Code and Classes Documentations

The source code is publicly available in the VTK source code. The files are located in the [Interaction/Widgets](#) folder:

- `vtk3DCursorWidget.h`
- `vtk3DCursorWidget.cxx`
- `vtk3DCursorRepresentation.h`
- `vtk3DCursorRepresentation.cxx`

The developer references are here:

<https://vtk.org/doc/nightly/html/classvtk3DCursorWidget.html>

<https://vtk.org/doc/nightly/html/classvtk3DCursorRepresentation.html>

Overall Design

As stated in the [VTK Book](#), the rendering pipeline of VTK consists of *representation* of data which transforms a computational form into a graphical form. Please refer to [Chapter 4](#) of the VTK Book for details.

In the VTK world, the user interacts and manipulates data through *actors*. Hence the 3D Stereo cursor is implemented at this level. Being an actor itself, the widget will be rendered in stereo like any other actor.

It consists of 2 elements: the 3D cursor widget and its associated representation. The `vtk3DCursorWidget` performs events handling and forwards them as actions to the representation accordingly. The `vtk3DCursorRepresentation` class performs the actions, i.e. the placement of the cursor actor in the scene, given the display position of the mouse.

Then at the computation level, the placement of the cursor uses a [Picker](#) which is in charge to compute the intersection between the projection of the 2D mouse cursor and the object underneath. VTK provides several pickers each dedicated to specific usage. For the 3D stereo cursor, the implementation uses an [Hardware Picker](#), which triggers a dedicated OpenGL rendering to retrieve the hovered object, combined with ray casting to compute the intersection.

The overall flow is described in figure 2.

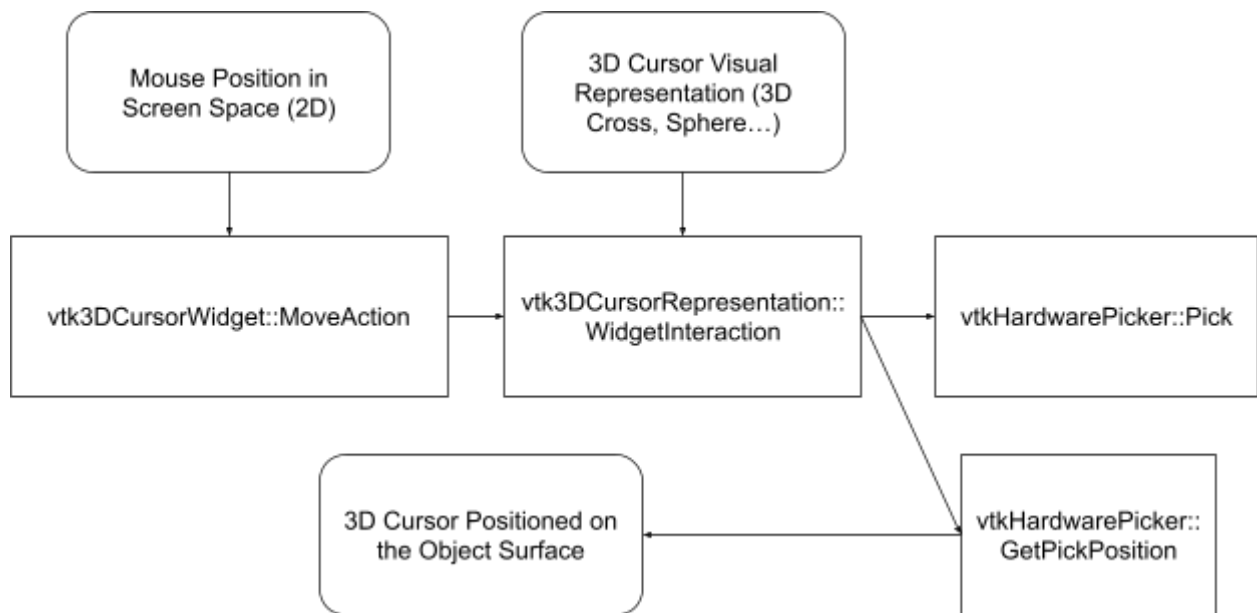


Figure 2. The high level description of the workflow with the involved objects: `vtk3DCursorWidget`, `vtk3DCursorRepresentation` and `vtkHardwarePicker`.

Usage in a VTK-based Application

The basic usage of the 3D stereo cursor happens at the render window interactor level.

The steps are:

- Create a `vtk3DCursorWidget`
- Set the current interactor to the 3D Cursor Widget
- Activate the Cursor (method "On()")
- Hide the 2D cursor of the interactor

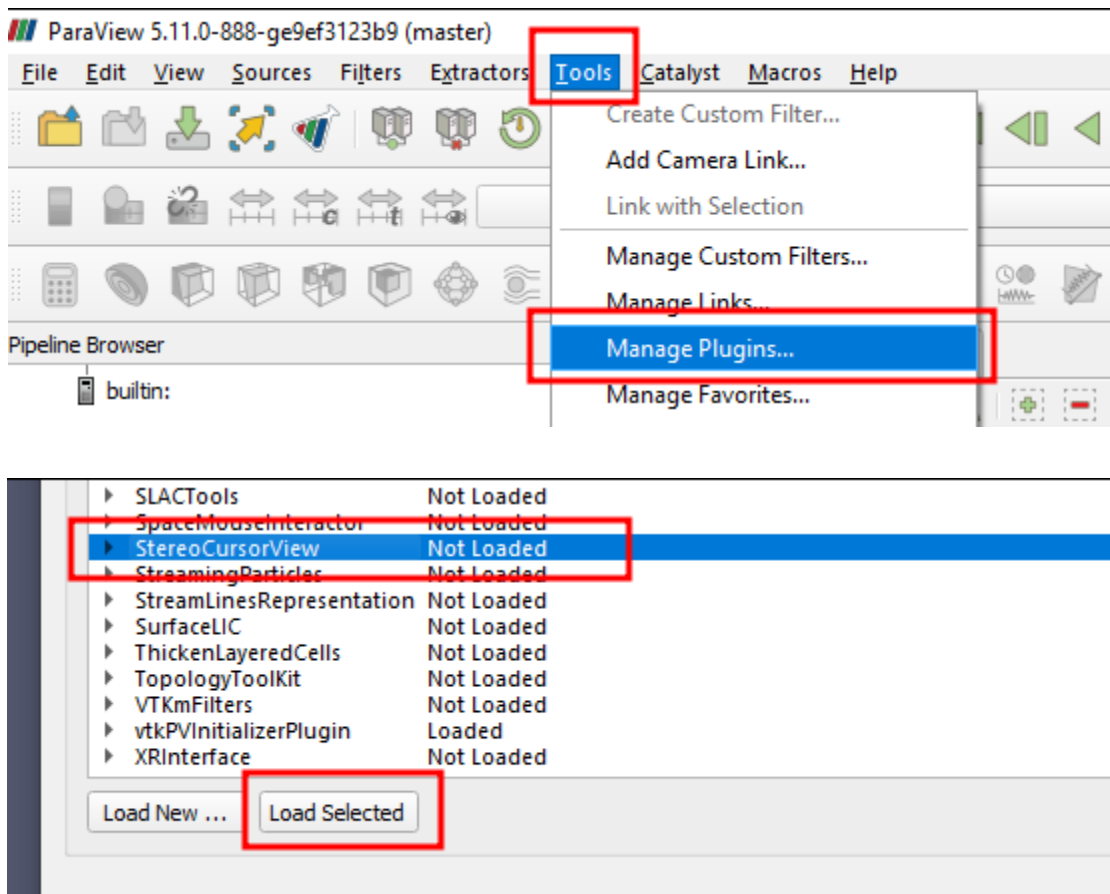
Corresponding C++ code:

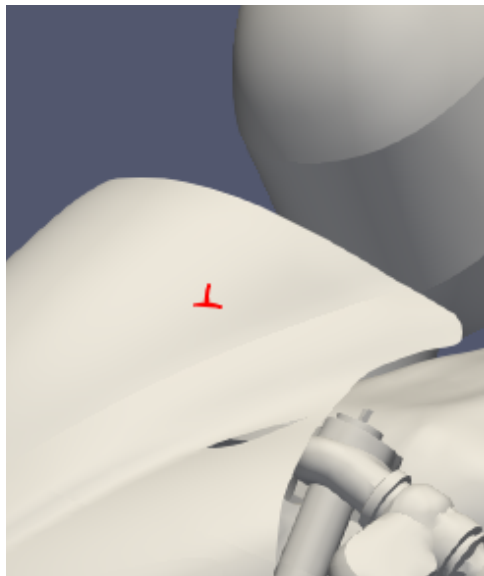
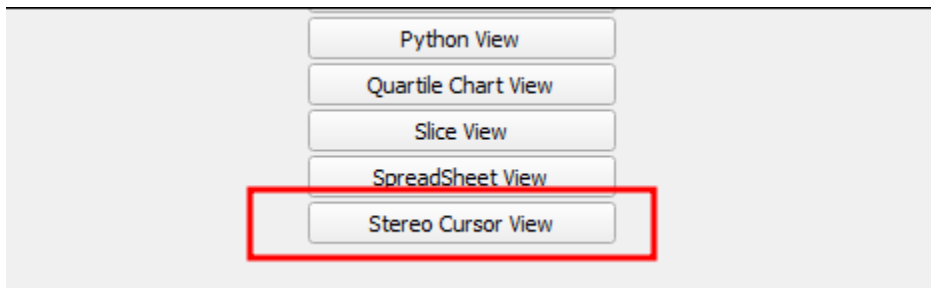
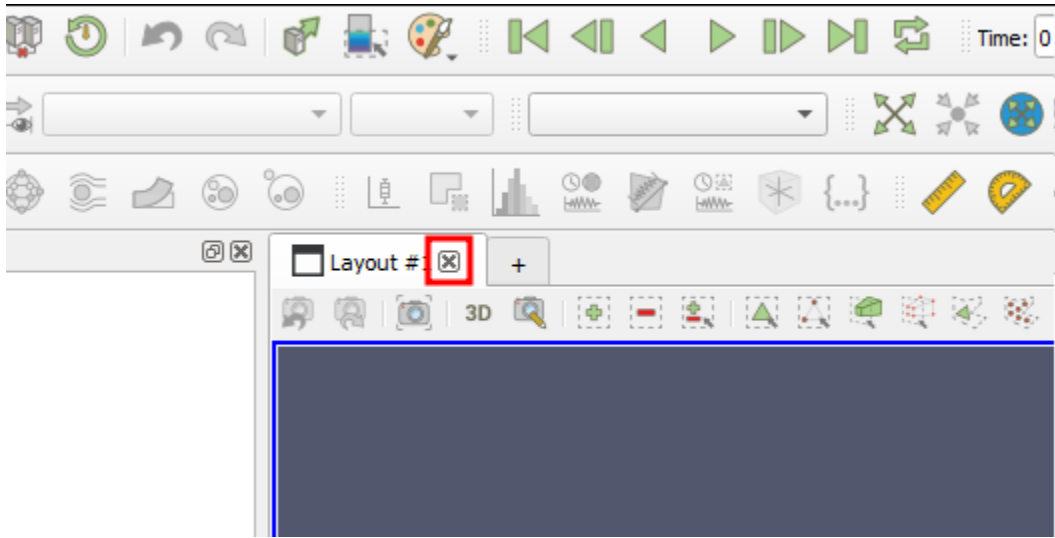
```
vtkNew<vtk3DCursorWidget> cursor;  
  
cursor->SetInteractor(renderWindowInteractor);  
  
cursor->On();  
  
renderWindowInteractor->HideCursor();
```

Integration Examples

ParaView

ParaView from version 5.12, delivers a dedicated plugin named StereoCursorView to display the 3D Stereo Cursor in a dedicated view:



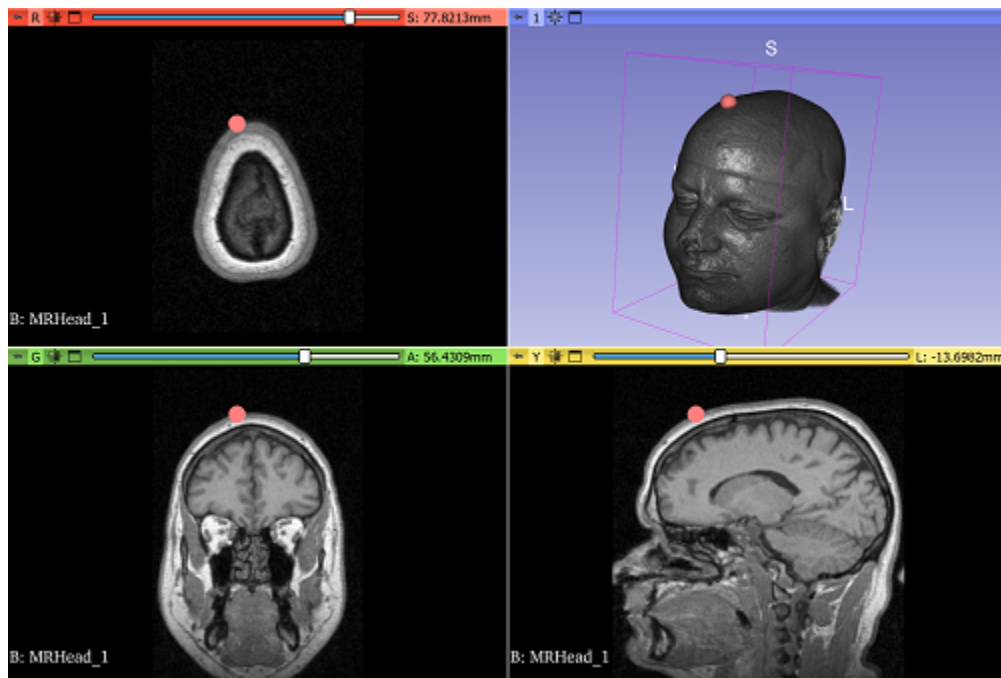
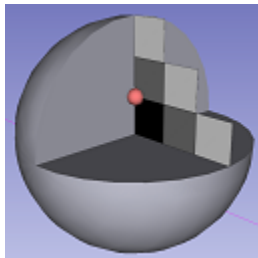


The ParaView plugin uses the VTK C++ API. Source code is here:

<https://gitlab.kitware.com/paraview/paraview/-/tree/master/Plugins/StereoCursorView>

3D Slicer

3D Slicer integrates the 3D mouse Cursor via a dedicated extension.



This extension replaces the application mouse pointer with a fiducial node that is directly rendered in the scene. The virtual cursor node is projected onto the surface or volume being hovered, or in the camera focal plane if no intersection is found.

It provides a single module in the "Utilities" category. When selecting the VirtualCursor module, the application mouse pointer is automatically replaced by a fiducial node when hovering the render views. The virtual cursor can be disabled at any time by unchecking the corresponding checkbox in the module's widget. A slider is available as well in order to control the size of the cursor.

The 3D Slicer extension uses the VTK Python API. Source code is here:

<https://github.com/KitwareMedical/SlicerVirtualMouseCursor>