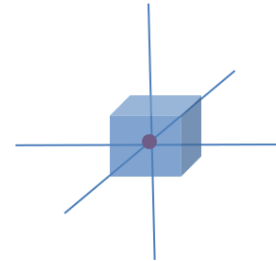


Subject: Stereoscopic Mouse Cursor (SMC)
Version: **03.3**
Date: Dec. 21, 2022



The Stereoscopic Mouse Cursor (SMC) Concept and Realization

Necessary for productive work in 3D-Stereo environments
Enables the best user experience for 3D-Data interaction

Introduction

The intention of Schneider Digital (SDIG) is to promote the implementation of a stereoscopic mouse-cursor (SMC) to take the stereoscopic viewing experience for all 3D-stereo display systems to the next level – the addition of a new generation of 3D object interaction capabilities.

Schneider Digital (SDIG) has been selling various stereoscopic display systems for the desktop and the large-format market for the past 18 years. As a hardware vendor, SDIG is 100% software-neutral and has since established an extensive network, connecting to all major international software vendors (ISVs).

Being a specialist for stereoscopic 3D display systems, SDIG is motivated by the profound knowledge of the already existing and potential, future benefits associated with working comfortably in a true 3D-stereoscopic, digital model environment. We live in a 3D world, which we intuitively understand in its true dimensions due to the undisputed evolutionary benefits of being born with two eyes. At just a few months of age, we can also judge what is within our reach and how we can use our arms and hands to get a hold of an object and interact with it. In this Whitepaper, we simply argue that all technical components are well in place to extend our natural 3D World to the professional desktop, allowing users with any (!) stereoscopic viewing system to use both eyes (!), perceive true depth in 3D model-space stereoscopically and interact intuitively through the well-familiar two-button mouse with any CAD, GIS, 3D-mesh models, LiDAR point cloud or any other 3D object environment.

Compared to the standard 2.5D 'flat-panel' work environments, i.e. the perspective viewing of 3D datasets without true depth perception, the stereoscopic mouse cursor (SMC) in combination with any stereoscopic display system opens up a host of new 3D model interaction capabilities for professional users. Similar interactive capabilities to reach out and interact with objects were previously available only in immersive VR gaming environments, necessitating the (mostly short-term) use of head-mounted devices (HMDs). Despite the challenges, commonly associated with the use of HMDs, VR game interaction in 3D model space proved to be the key to the commercial success of the VR gaming industry.

The provided information and materials should give experts for OpenGL / VULKAN and DirectX the necessary background and documentation, combined with some impressive sample applications, to achieve SMC implementation success within a short period of time. This is a working document under constant revision and an open-source effort by several parties, merely coordinated by Schneider Digital. Contributions and corrections to this document are most welcome.

General Background Introduction

Schneider Digital is currently collecting and distributing a set of SMC-related materials, such as:

- White Papers;
- Open-source Documents;
- Source Code Examples;
- Open-Source Demo Applications.

All of the above are related to the subjects '*3D-Stereo Visualization*' and '*Stereoscopic Mouse Cursor (SMC)*', regardless of the specific display technology that is being used with it. The SMC implementation is valid for all stereoscopic displays, e. g. autostereoscopic, line-polarized or 120Hz-stereo flat panels with and without active/passive glasses, stereoscopic Laser projection systems, head-mounted VR and AR devices, LED walls and caves, including all future stereoscopic display technologies.

The material presented also pursues a cross-platform approach and should have desktop support for Windows, Linux, macOS, and mobile support for Android and iOS, and future operating system versions.

Stereoscopic Mouse Cursor Introduction

For any 3D-stereo enabled application, the best user experience depends very much on:

- a) How deep can we involve the user in a 3D environment – **comfortably (!)** ?
- b) How easy can the user reach every surface, every point and edge, and every 2D-pixel or 3D-voxel, - **intuitively (!)** - with just a standard, two-button mouse ?

Remarks:

It is not an option nor objective to implement special 3D mouse or other 3D controller devices. The proposed SMR works with any standard mouse, even with a laptop touchpad.

What is the practical benefit of the SMC?

Today, most 3D-stereo enabled applications use the normal 2D operating system cursor also for 3D-stereo display scenarios. In this case, the cursor is visually hovering above the 3D model and needs to be carefully aligned with an object underneath, to correctly identify/select this element. It is obvious that this is not an efficient and rather error-prone solution, requiring a careful view alignment of the object to the distant 2D cursor.

In contrast, the SMC works similar to the light dot of a laser pointer, which also 'attaches' itself to any 3D object it encounters in its path. With an automatically scalable and/or configurable display of this laser 'dot' as crosshairs or any other semi-transparent marker, the user can then zoom-in or zoom-out, shift or rotate the view around this anchor point on the object. This is much faster and more efficient than manipulating a 3D-stereo view by a distant and detached 2D cursor. A relevant keyword for this comparison is '3D-Picking', vs. '2D / 2.5D selection'.

3D-stereo enabled applications with SMC functionality to test

The following sample applications can be downloaded directly from the Schneider Digital website:

➤ **Aardworx stereo**

New **Tomography** software with a nice SMC integration! Enables even the picking **inside a volume** by simply holding down the **control key** on your keyboard. Very nice demo functionality!

- **CAD Assistant** is **freeware**, so you don't have to get a license and it works Plug & Play with our 3D PluraView or every other stereoscopic visualization device.
CAD SAMPLE-File for testing the functionality in CAD Assistant.

- **TerraStereo** is **freeware**, so also for this application you don't have to get a license and it works Plug & Play with our 3D PluraView or every other stereoscopic visualization device.
PointCloud SAMPLE-File for testing the functionality in Terra-STEREO.

Acute3D has not implemented the SMC yet, but Acute3D has implemented one of the BEST stereo cameras, i.e. stereo viewport integration!

- **Acute3D viewer** is **freeware**, so you don't have to get a license and it works Plug & Play with our 3D PluraView or every other stereoscopic visualization device.
Photogrammetry mesh SAMPLE-File for testing the functionality in Acute3D (now ContextCapture).

The following section has kindly been provided by:

Aardworx

harald.steinlechner@aardworx.at

Introduction (by Schneider Digital)

Aardworx is a highly innovative technology provider, based in Vienna / Austria. Their team of developers is dedicated to the volumetric rendering of complex 3D sensor data and Computer Tomography (CT) scans, the efficient and fast display of point clouds of any size and the general visualization/streaming of data across the Internet to enable remote collaboration even with the most complex and voluminous datasets.

SMC Rendering at object depth (by Aardworx)

Surfaces: can be modeled using polygons or NURBS. There are different ways of displaying them:

- Solid/Shaded (*1): Regarding cursors, the straightforward method works where you use the pixel under the 2D cursor to calculate the 3D position.
- Wireframe: The lines are typically very thin. For normal picking on a 2D screen, you often have to make the line "thicker" to hit it with the normal mouse. Most applications should already have this option. For stereo-picking there is another issue:
 - If you don't hit a line - you have to place the cursor at some predefined depth; this way, the cursor does not jump permanently between different depths. This is similar to the 3D picking problem with point clouds (see below).
 - One possibility is to render lines even "thicker" - the better solution is to use a similar picking strategy as (*1), but still display it as a wireframe. This way the cursor stays on position even next to lines, and follows the surface described by the line.

Point clouds: Similar to wireframes rendering - the cursor may jump when no point is located under the 3D cursor. There are two solutions:

- When there is no hit, the cursor could just freeze at the last successful hit, which is feasible only for dense point clouds and/or low magnification factors.
- The cursor can still be moved freely, but retains its depth location.

Volumetric data: Volumetric data is a particularly interesting case because a model may consist of many surfaces stacked inside each other or even does not consist of any surfaces at all. A solution here is to consider all surfaces under the SMC and allow the user to switch between e.g. snap to the first surface in the viewing direction, then the second one and so on. This interaction works especially well in stereo.

Integrating Picking with 3D Views

Inside the 3D-stereo view the 2D cursor of the operating system should be disabled. The rationale behind this is, that the stereo cursor is composed by both the left and the right eye display. Zero disparity corresponds to a

certain depth in 3D space. As a result, objects and the pointer onto those objects appears at different depth. This breaks immersion completely for most people.

The easiest solution is to explicitly render a cursor at the correct depth manually. The simplest solution is to use a textured quad with the texture of a cursor.

Which depth is the correct display depth for the SMC? As an actual object in 3D, the cursor should have the same display depth as the 3D object the user is pointing at. If your rendering engine or scene data structure supports ray picking, clearly a pick-ray could be computed using either the left, or the right view/projection matrices. With this approach, the 3D coordinate of the hit point can be used to render the cursor.

Another option is to use depth readback from a framebuffer. In graphics applications which render directly to the framebuffer (i.e. no deferred pipeline), a simple **glReadPixels** suffices to read back the depth value. It shall be noted, that this involves the decision which framebuffer to read the depth values from. For a partially visible object, for instance at the screens' edge or one of the corners, this leads to unexpected picking results (e.g. an object is only visible in one eye). Also, please note that when using either left, or right eye for "unprojection", the projected position will have an offset respectively to the system cursor. Since the system cursor should be disabled for other reasons as well, this is not that problematic, yet the cursor might jump, while moving around. While this could be fixed by rendering a third framebuffer using the center projection, often just using just the left or right framebuffer suffices for acceptably precise interactions. Depending on the conventions used in the graphics API and the matrices, the pixel can be unprojected using:

```
// compute normalized device coordinate
```

```
ndc = (mousePos.X / viewportSize.X, mousePos.Y / viewportSize.Y, d * 2.0 - 1.0, depth, 1.0)
```

```
p = viewProj.Inverse * ndc // backproject
```

```
worldPos = p.XYZ / p.W // perspective division
```

For deferred pipelines, reading back could for example be done using **glGetTexImage**. While the download introduces a CPU synchronization point, often this is not a problem in professional applications.

Clearly there are better, less conventional approaches to 3D cursors, depending on the data. For example, for polygonal 3D geometry a virtual sphere projected onto the surface to create a footprint can provide good depth-perception (see Figure 1).



Figure 1: "Footprint Cursor" combined with the standard windows cursor Pro3D [7]. This approach is easily portable to 3D

Such effects can be achieved solely in the pixel shader. Given the world space cursor position, the distance of the fragment position to the cursor position can be computed and thus used for colorizing the pixel. In order not to suffer from shader aliasing, the 'smoothstep' [9] function can be used to model smooth thresholds.

A similar technique can be used for volumetric data. Figure 2 shows the selection of a pore in 3D volumetric data. Instead of having a pointer-like cursor, pores “under” the cursor are colored in red. This technique is very similar to the previously mentioned approach. Given the cursor’s world-space coordinates, the distance of the voxel to be shaded and the cursor can be computed directly in the shader in order to colorize voxels within a distance to the cursor position.



Figure 2: Billboard (3D) with selected pore of casting part in Aardworx Volume Renderer [8]. For 3D viewing the billboard should be placed at proper depth.

Another option is to use polygonal objects instead of a 2D cursor or modified shading (see Figure 3).

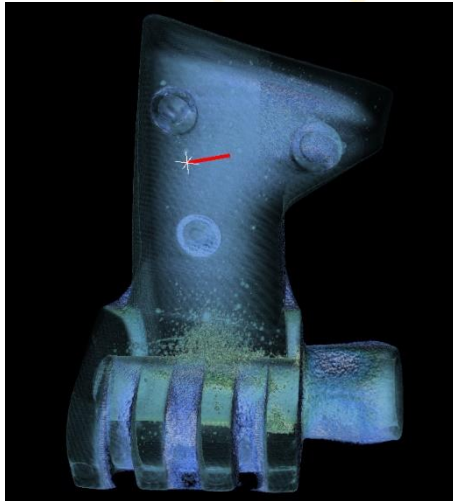


Figure 3: Zink casting part in 3D. Cursor Cross, rendered at correct depth and measurement line in Aardworx Volume Renderer [8]. This technique works well for 3D display settings.

Advanced Topics

When using the readback technique, clearly the back-projected world space coordinate always exactly matches the pixel under the mouse. However, when using a cursor representation larger than a single pixel, the cursor might get occluded in the surrounding when depth testing is enabled. The situation is illustrated in Figure 4. In their study Schemali and Eisemann find that rotating the cursor and using a circle pointer improves usability.

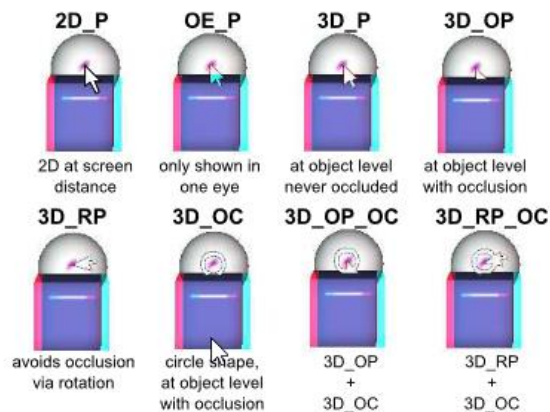


Figure 4 Problems of cursors as analyzed by Schemali and Eisemann [10]

Remarks (LM): Please use ANAGLYPH-Glasses (Red-Cyan) for evaluating Figure 4.

Beyond picking and mouse interactions for pointing and selection, there is a lot of research in the field of virtual reality, tackling interaction by using virtual reality devices and tracking. Natalia et al. [13] for example present a study on how to integrate 2D and 3D interactions for modelling applications.

So far, we have identified the following relevant documentation, which are referring to the SMC:

- | [Using Ray Casting To Move 3D Objects \(ogldev.org\)](http://ogldev.org)
- | [3D Picking Tutorial-29 \(ogldev.org\)](http://ogldev.org)
- | [Stereo 3D Mouse Cursor: A Method for Interaction with 3D Objects in a Stereoscopic Virtual 3D Space \(hindawi.com\)](http://hindawi.com)
- | [Design and Evaluation of Mouse Cursors in a Stereoscopic Desktop Environment \(archives-ouvertes.fr\)](http://archives-ouvertes.fr)
- | [Stereo 3D mouse \(S3D-Mouse\): measuring ground truth for medical data in a virtual 3D space - PubMed \(nih.gov\)](http://nih.gov)
- | [Depth Cues and Mouse-Based 3D Target Selection \(carleton.ca\)](http://carleton.ca)
- | [HybridSpace: Integrating 3D Freehand Input and Stereo Viewing into Traditional Desktop Applications \(autodesk.net\)](http://autodesk.net)
- | [Stereo pair selection, measurements and navigation: Metashape Helpdesk Portal \(agisoft.com\)](http://agisoft.com)

References

- [1] Grasberger, Introduction to Stereo Rendering,
https://www.cg.tuwien.ac.at/research/publications/2008/Grasberger_2008_ISR/Grasberger_2008_ISR-Paper.pdf
- [2] https://github.com/ieheid/quad-buffered-stereo/blob/fe72d5813500df751a899b86c30826fae9186b0b/jpers_stereo.c#L497
- [3] https://www.khronos.org/opengl/wiki/Geometry_Shader#Layered_rendering
- [4] https://www.khronos.org/registry/OpenGL/extensions/OVR/OVR_multiview.txt
- [5] <https://www.khronos.org/registry/vulkan/specs/1.3-extensions/html/vkspec.html#VkPhysicalDeviceExternalImageFormatInfo>
- [6] <https://github.com/aardvark-community/aardium>
- [7] <https://pro3d.space/>
- [8] <https://www.aardworx.at/>
- [9] <https://www.khronos.org/registry/OpenGL-Refpages/gl4/html/smoothstep.xhtml>
- [10] *Design and Evaluation of Mouse Cursors in a Stereoscopic Desktop Environment*, Schemali and Eisemann,
<https://hal.archives-ouvertes.fr/hal-01026538/document>
- [11] Teather, Stuerzlinger, Depth Cues and Mouse-Based 3D Target Selection
https://www.csit.carleton.ca/~rteather/pdfs/SUI2014_poster1.pdf
- [12] Azari, Cheng, Basu, Stereo 3D Mouse Cursor: A Method for Interaction with 3D Objects in a Stereoscopic_Virtual 3D Space, <https://www.hindawi.com/journals/ijdm/2010/419493/>
- [13] Bogdan, Grossman, Fitzmaurice, HybridSpace: Integrating 3D Freehand Input and Stereo Viewing into Traditional Desktop Application <https://damassets.autodesk.net/content/dam/autodesk/research/publications-assets/pdf/hybridSPACE-integrating-3d-freehand.pdf>

The following section has kindly been provided by:

TEHNOGIS

octavian.balota@tehnogis.ro

Introduction (by Schneider Digital)

TEHNOGIS in Bucharest was established in 1995 as a highly innovative geospatial data provider, realizing among many other project achievements the first airborne LiDAR survey in Romania in 2001 and the first aerial imaging of Bucharest in 2006. Due to their pioneering activities in Geodesy, Photogrammetry, Cadastre and urban mapping, they have developed numerous internal software applications, enabling complex workflows to be executed effectively and efficiently. This includes their own development work, implementing 3D data interaction workflows with integrated stereoscopic mouse cursor capabilities.

The concept of an Intelligent 3D Stereo Mouse Cursor to be used in stereo view applications (by TEHNOGIS)

From the **visibility** point of view, the Stereo Mouse Cursor (SMC) should have a maximum visibility regardless of the 3D data you work with, or the level of Zoom you are using.

From the **scoring accuracy** point of view, the SMC should have the ability to position itself with maximum precision on any kind of 3D data, such as stereo images, point clouds, 3D meshes, 3D CAD objects, etc.

From the **behavior** point of view, the SMC should exhibit such a behavior that it allows the improvement of both the scoring precision and cursor visibility during work without the need for special setting adjustments. The SMC should react to the specific work environment and application. The SMC shape, colors and size should adapt to the work environment for an optimal visibility and accuracy.

The main characteristics of an SMC are the shape and the color.

Proposals for shape availability and expression:

- 1) The basic shape is the point. It is the simplest shape that will be part of all other shapes.
- 2) Simple cross (traditional). This is 2D shape but moving in a 3D environment
- 3) 3D cross (3D coordinate axes). This could be a representation of only the axes in the 3 positive (+) directions or combined with the negative directions, originating in the intersection point of the axes
- 4) Simple circle (traditional). This is a 2D shape but moving in a 3D environment.
- 5) Sphere
- 6) Cub
- 7) Arrow

Proposal for shape parameters to improve visibility and accuracy:

- 1) Color with adjustable intensity (white, black, green, red, blue, pink)
- 2) Visibility (values on/off)
- 3) Pulsation (values on/off, frequency - if 'on')
- 4) Orientation (not valid for point and sphere); Values(on/off) Coordinate orientation - if 'on'
- 5) Size (adjustable with (+) and (-) on keyboard)
- 6) Planimetric speed depending on X, Y step values (adjustable with PgUp and PgDn)
- 7) Vertical speed depending on Z step values (ArrowUp and ArrowDn)

In general, all these SMC parameters should be easily adjustable during work.

Class Diagram SMC

