Nordic Neuro Lab

PsychoPy

VisualSystem HD

Using PsychoPy with the VisualSystem HD



0 Table of Contents

1	Introduction	3
2	Hardware and display setup	3
2.1	Recommended hardware	3
3	Installation	4
4	PsychoPy experiments on the VSHD	4
4.1	Run and develop scripts	4
4.2	Using the VisualSystemHD class	5
4.3	Interpupillary distance adjustment	5
4.4	Distortion correction	6
4.5	Reducing artifacts	7
5	PsychoPy Demo	7
5.1	2D example	8
5.2	3D example with IPD adjustment	8
5.3	Distortion correction example	9
6	References	10

Definitions

VSHD	VisualSystem HD
IPD	Interpupillary Distance
MSAA	Multisampling Anti-Aliasing
MRI	Magnetic Resonance Imaging
IDE	Integrated Development Environment

1 Introduction

PsychoPy is a free cross-platform package which can be used to run experiments for behavioral sciences [1]. This document covers the use of PsychoPy on Windows. PsychoPy is written in Python and is open source. It can be used to present stimuli and measure responses. NordicNeuroLab's VisualSystem HD (VSHD) makes it possible to present stimuli inside an MRI scanner. Within PsychoPy, experiments can be built using the "Builder" GUI, or as a Python script using the integrated code editor ("Coder"). Finally, the "Runner" window allows the user to manage and run experiments. We will focus on the Coder and Runner windows when working with experiments for the VSHD.

In this guide we will look at both the hardware display setup and software elements which allow PsychoPy to be used with the VSHD. We will also go through how to use PsychoPy and show examples on how to implement scripts to create visual stimuli for the VSHD both in 2D and 3D. The geometric distortion of images in VSHD will be discussed as well as how to adjust for this problem using predistortion in PsychoPy. In addition, we will look at how to decrease artifacts caused by predistortion, as well as adjustment of 3D rendering according to the subject's interpupillary distance. This manual assumes a basic understanding of both Python and PsychoPy.

2 Hardware and display setup

As the VSHD goggles comprise two independent displays, one for each eye, you can show either the same (monoscopic) or different (stereoscopic) images on each display. When using PsychoPy with the VSHD, two images – either the same or different – are sent to the displays. To achieve this a stereoscopic setup is used, more specifically the Setup 2S Combined provided in our Display Setup for VSHD guide [2]. Note that the image will appear as one when looking into the goggles, but when working on a monitor you will see two images as showcased in section 5.

If only 2D stimuli are to be displayed, it is possible to use the 1M monoscopic setup described in the VSHD guide [2]. However, then the VSHD specific class and its parameters discussed in this guide cannot be used as it only works with a stereoscopic setup. For 2D stimuli, please refer to the Window class described on the PsychoPy website [3].

When working with PsychoPy and the VSHD, windows can occasionally open on the wrong display. If this happens select the window you want to move on the taskbar and use Windows + Shift + Left/Right Arrow as described in Section 4.4 of the VSHD guide [2]. Please note that this function should not be used to move a running PsychoPy stimuli from an incorrect display onto the VSHD display, as it may result in a wrongly proportioned image. The correct display for running the stimuli can be chosen with PsychoPy, as shown in section 4.1.

2.1 Recommended hardware

To use PsychoPy it is required that your computer has a graphics card that supports OpenGL, ideally OpenGL version 2.0 or higher.

3 Installation

PsychoPy can be installed on your computer using the link below. To write and run scripts you need to install Python as well. Python 3.8 is the currently recommended version by PsychoPy for the most stable results, but newer versions will also work.

- **PsychoPy:** https://www.psychopy.org/download.html
- **Python:** https://www.python.org/downloads/

The demo scripts can be installed from the GitHub repository given in the link below:

- **Example scripts:** https://github.com/nordicneurolab/vshd-psychopy-example

As of June 2022, the 3D Demo provided will not run on the newest version of PsychoPy. A fix for this is being developed by PsychoPy for their next major release. The 3D example will run on version 2021.1.0, however. This version can be downloaded from GitHub using the following link:

- **PsychoPy 2021.1.0:** https://github.com/psychopy/psychopy/releases?page=2

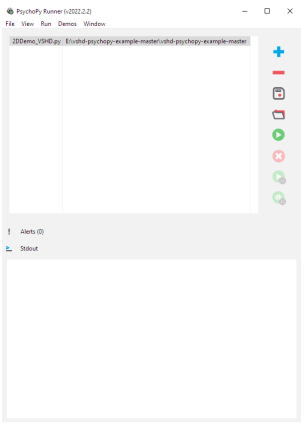
When using the link above, click on Assets and then click on StandalonePsychoPy-2021.1.0, and choose the version corresponding to your OS. If you already have a version of PsychoPy on your computer, it is highly recommended that you uninstall it first.

4 PsychoPy experiments on the VSHD

4.1 Run and develop scripts

Running a script in PsychoPy can be done in one of two ways. You can use the Runner window (Figure 1a) by pressing the blue plus button and locating your script. The second option is to use the Coder window (Figure 1b) to open an existing python file by pressing File and then Open. In the Runner window you can also open multiple scripts and then choose which one to run using a left mouse click. The scripts are run by pressing the green play button in the Coder window or Runner window, depending on which one you are using. The experiment can be stopped by pressing the red cross in the Runner window.

a



b

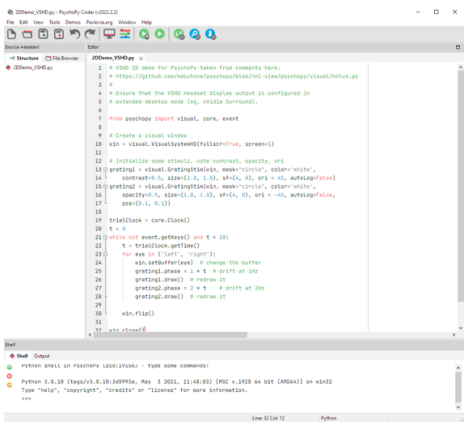


Figure 1: (a) Runner and (b) Coder window in PsychoPy GUI

If you want to develop your own scripts with PsychoPy you can either write within the PsychoPy Coder window or choose to use your own Integrated Development Environment (IDE). When creating a script with PsychoPy for the VSHD, use the VisualSystemHD class. More detailed information on the class can be found in the class documentation on the PsychoPy website [1].

4.2 Using the VisualSystemHD class

The VisualSystemHD class takes multiple arguments. Two important ones to use are the full screen parameter (fullscr =) and screen parameter (screen =). Set the fullscr parameter to True to make sure that the stimuli are shown in full screen. The screen parameter decides which monitor to display the visuals on. However, PsychoPy and the OS may number your screens differently. Try to set the screen parameter equal to zero and increase the parameter according to the number of screens connected. For instance, if you have one monitor in addition to VSHD try first "screen = 0", and then "screen=1". Examples of this are shown in the demo scripts provided and in Figure 2.

```
from psychopy import visual, core, event

window = visual.VisualSystemHD(fullscr=True, screen = 0)
```

Figure 2: Initializing a PsychoPy window with the VisualSystemHD class

When using the VisualSystemHD class to show stimuli, a buffer is set for each eye, or more precisely a buffer is set for each side of the selected screen. You may see that the class takes a monoscopic Boolean argument, but this will only determine whether the same image is drawn to both eye buffers. Setting monoscopic to True is recommended if you only intend to display 2D stimuli about the center of the display as it uses a less memory intensive rendering pipeline. By default, stereoscopic rendering is used which is required for showing 3D stimuli correctly. When using stereoscopic rendering we will have to change the buffer between left and right eye and draw the stimuli separately for each frame to achieve a 3D effect.

4.3 Interpupillary distance adjustment

Displaying 3D visual stimuli using the VSHD requires the Interpupillary Distance (IPD) between the left and right cameras in the virtual scene to be placed at the same distance as the IPD of the subject viewing the scene. The VSHD can be physically adjusted to the subject's IPD by turning the IPD knobs. When the VSHD is adjusted to the subject's IPD, it is necessary to find the separation by locating the centers of the lenses and measuring the distance between them (Figure 3).



Figure 3: IPD on human eyes and VSHD

Adjustment for the IPD in PsychoPy scripts is done by adjusting the eyeOffset parameter. When changing between your two eye buffers, the eyeOffset should be set to half of the interocular separation. A negative offset value is used for the left eye buffer, and positive for the right eye buffer. The correct perspective is applied to the 3D rendering using setPerspectiveView. An example on how to use the eyeOffset for IPD adjustment is shown in the 3D example script in the provided repository and in Figure 4 below.

```
ipd = 6.0
while 1:

    for eye in ("left", "right"):

        win.setBuffer(eye)
        if eye == "left":
            win.eyeOffset = -ipd / 2.0
        else:
            win.eyeOffset = ipd / 2.0

        win.setPerspectiveView()
```

Figure 4: How to set the eyeoffset in PsychoPy. 'Window' refers to a VisualSystemHD window as created in Figure 2

4.4 Distortion correction

The VSHD has a small but unavoidable pincushion distortion effect due to an increasing magnification away from the optical axis. This results in lines in the image curving outwards from the center (Figure 5c). The inverse effect, or barrel distortion (Figure 5b), can be applied to the image before it is sent to the VSHD to compensate for this effect, resulting in an undistorted image (Figure 5a) [4].

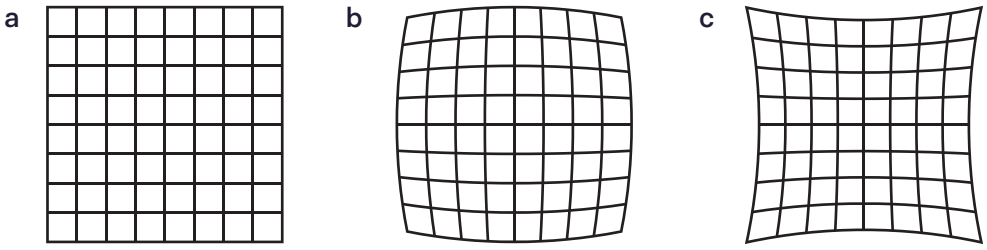


Figure 5: (a) Object with (b) barrel and (c) pincushion distortion [5]

The VisualSystemHD class offers an adjustment of the applied barrel predistortion. This can be done by setting a distortion coefficient ("VisualSystemHD.distCoef") equal to some floating-point number. The coefficient is set to -0.02 by default, but users must empirically verify whether this value is correct. Even though the distortion will be very similar for all VSHD displays, there may be small variations between the individual VSHD units, as well as for different diopter settings. A script that displays a square grid can be used to tune the coefficient to the value which results in the most correct presentation of the grid to the subject. An example of this script and how to find a correct value is given in section 5.3.

In Figure 6, the left-hand part of the screen has a default amount of pre barrel distortion applied, whereas the right part has no pre distortion. When looking at the stimuli in the VSHD, the left image will appear normal with close to no distortion. The right image will have some observable pincushion distortion. However, if you are to display stimuli where the content is near the center of the screen, the effect is minimal, and the user should consider whether pre-distortion is necessary.

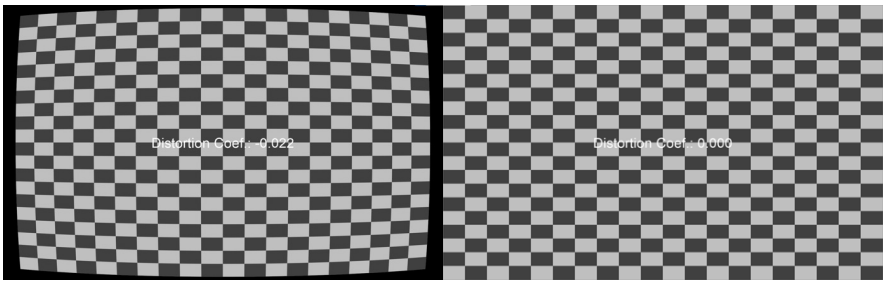


Figure 6: Stimuli from the distortion demo, with and without pre-distortion

4.5 Reducing artifacts

Applying the barrel predistortion algorithm may result in a slightly lower visual fidelity due to artifacts around the edge of the distorted image. Aliasing will result in perceptible staircases in place of the straight lines around the edges of the image [6]. The VisualSystemHD class can apply a multi-sampling anti-aliasing (MSAA) technique in the predistortion algorithm by setting multiSample to True and numSamples equal to a chosen integer. This will reduce the artifact. An example on how to use these arguments is given in the Distortion Demo provided in the GitHub repository.

Figure 7 shows a checkerboard pattern from the Distortion Demo, run with 1 sample (Figure 7a) and 16 samples (Figure 7b). If you look closely, small white squares appear around the edges of the image when multisampling isn't used. When multisampling with 16 samples is applied, these white squares disappear, and the image is even around the edges as it should be.

a b

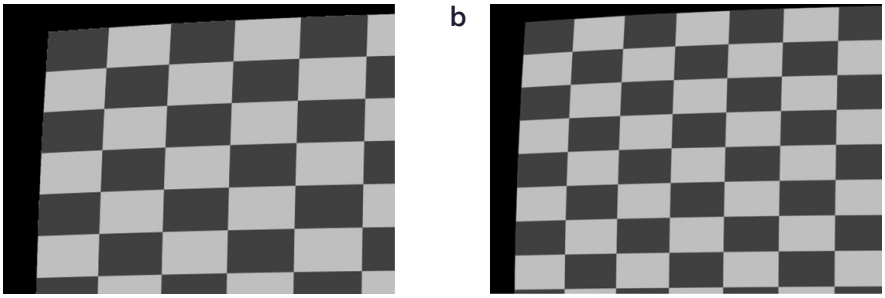


Figure 7: Multisampling in PsychoPy with 1 sample (a) and 16 samples (b)

5 PsychoPy Demo

The following examples will show how to use the PsychoPy VisualSystemHD classes. There are three scripts in the provided GitHub repository in section 3. The first example shows how to display a simple 2D animation while the second example illustrates a more advanced 3D animation. The last example provided shows how adjusting the barrel distortion will affect the image observed on the VSHD.

5.1 2D example

This script shows a moving grate in 2D. The same image is shown on both sides of the screen and the stereoscopic setup makes them appear as one for the user. The image sent from the computer to VSHD is shown in Figure 8.

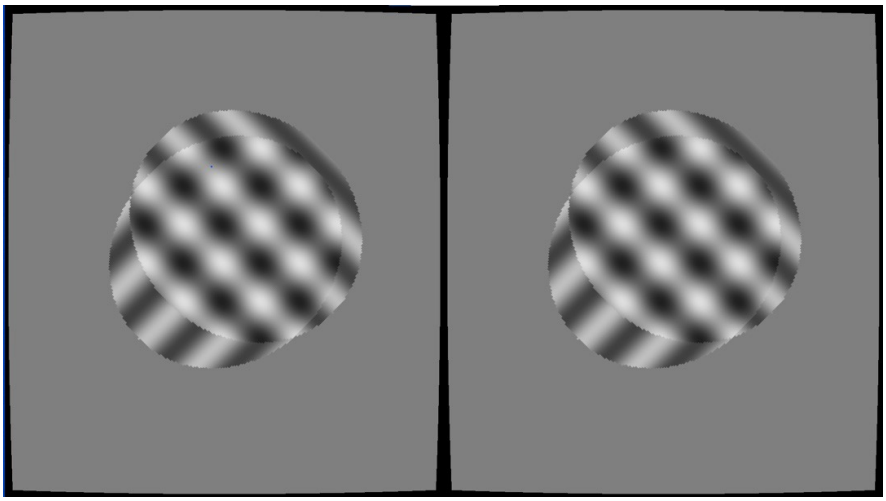


Figure 8: 2D demonstration in PsychoPy

The edges of the images can be a bit jagged; this is caused by the barrel predistortion. By using multisampling, this aliasing can be avoided as explained in section 4.5.

5.2 3D example with IPD adjustment

In this example we will look at how to display 3D images on the VSHD. It is based on the stim3d example that is provided under Demos in the PsychoPy Coder window. A spinning 3D cube is displayed on each side of the screen and the stereoscopic rendering will give a sense of depth for the subject. The image sent from the computer to the VSHD is shown in Figure 9.

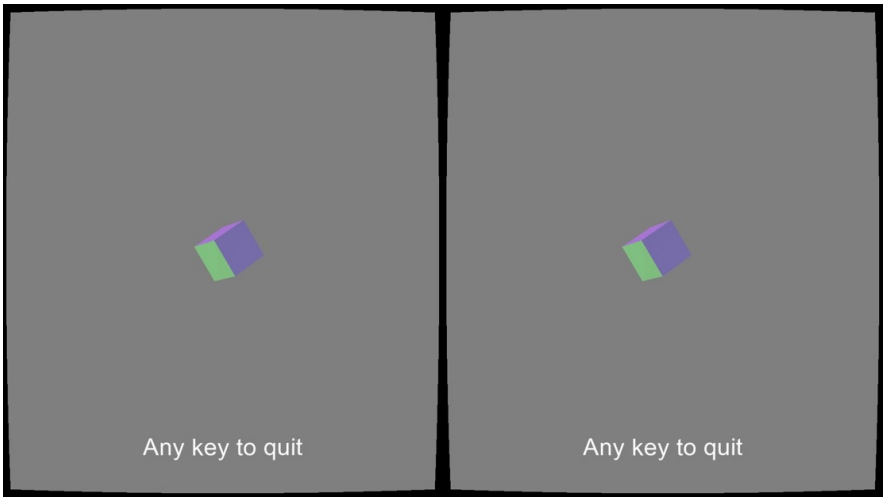


Figure 9: 3D demonstration in PsychoPy

The IPD needs to be set correctly to achieve the correct perception of the 3D cube. In this example, it was set to 6.0 cm which gave a good depth perception for the subject. Adjusting the IPD too high will result in the subject no longer observing the two cubes displayed as one, but two separate cubes.

5.3 Distortion correction example

This example shows a grid of black and white squares. The image sent from the computer is shown below in Figure 10, and this will appear as a single grid on the VSHD when set up correctly.

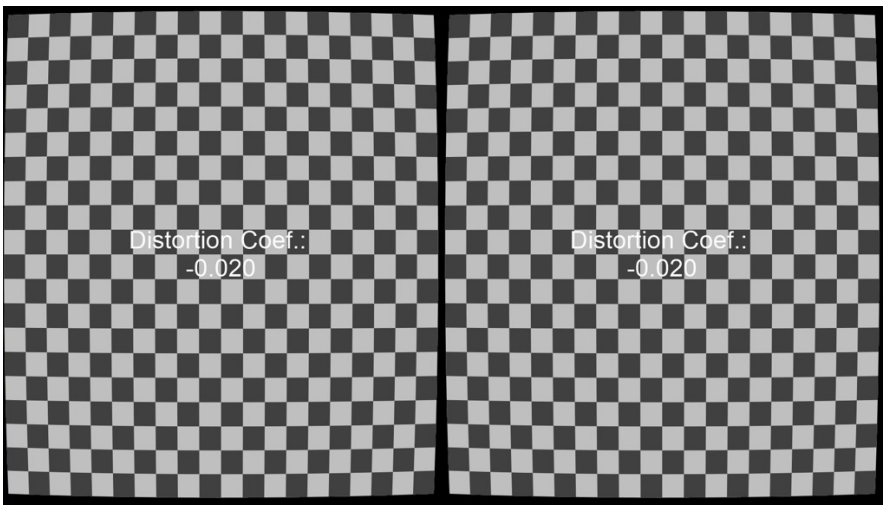


Figure 10: Distortion example for VSHD in PsychoPy

In the example script you can adjust the distortion coefficient by pressing w (increase) and s (decrease). When decreasing the coefficient, the user will see that the image will appear more rounded at the edges, showing a barrel distortion effect. By increasing the coefficient, and therefore reducing the predistortion effect, the lines in the grid will start to curve outwards from the center, and the inherent pincushion distortion will become visible. By using this script, the user can empirically find the correct coefficient to use when displaying stimuli using PsychoPy. The goal is, again, to make the grid as rectilinear as possible. In this example the distortion coefficient was tuned to -0.022 for the best result.

6 References

1. PsychoPy. VisualSystemHD [online] May 21, 2022. [Cited: June 21, 2022.]

   https://psychopy.org/api/visual/vshd.html

2. Display Setup for VisualSystem HD. NordicNeuroLab. [online]

   https://info.nordicneurolab.com/display-setup-for-vshd-01

3. [3] PsychoPy. Window [online]

   https://www.psychopy.org/api/visual/window.html

4. J. Tong, R.S. Allison, L.M. Wilcox. The impact of Radial Distortions in VR Headsets in Perceived Surface Slant [online] Dec 2019. [Cited: June 22, 2022]

   https://static1.squarespace.com/static/565e05cee4b01c87068e7984/t/6000a8ce822fd20bb4af6463/1610655950367/Tong_Allison_WIlcox_2019.pdf

5. Foundry Learn. Camera Matching [online] [Cited: June 21, 2022.]

   https://learn.foundry.com/modo/14.1/content/help/pages/rendering/camera_matching.html

6. S.M. LaValle. Chapter 7 Visual Rendering, Virtual Reality. [online] 2020. [Cited: June 23, 2022]

   http://lavalle.pl/vr/vrch7.pdf

Nordic Neuro Lab

CE

VisualSystem HD

Headquarters
NordicNeuroLab

Møllendalsveien 1
N-5009 Bergen
Norway

Regional Office
NordicNeuroLab, Inc.

234 W. Florida St., Suite 202
Milwaukee, WI 53204
USA

NL-100-0100-01