Nordic Neuro Lab

Psychtoolbox

VisualSystem HD

# Using Psychtoolbox with VSHD

VisualSystem HD

# 0 Index



# Definitions

| | |
|---|---|
| VSHD | VisualSystem HD |
| FOV | Field of View |
| IPD | Inter Pupil Distance |

# 1 Introduction

Psychtoolbox is a free set of Matlab/Octave functions developed for vision and neuroscience research (1). By incorporating low-level hardware control into the high-level interpreted language of Matlab/Octave, Psychtoolbox makes it easy to synthesize and show accurately controlled visual and auditory stimuli in a development-friendly environment. NordicNeuroLab's VisualSystem HD (VSHD) brings all this functionality into the field of fMRI by making it possible to show the stimuli in an MRI scanner.

We will first look at how to set up the hardware and software, including how to correct for the slight but unavoidable geometric distortion of the VSHD optical system. Four scripts are then provided with guidance to illustrate how to use Psychtoolbox with the VSHD. Serving as an indication of what Psychtoolbox can offer, the four scripts show how to display 2D, "fake 3D" and "real 3D" stimuli. The scripts are based on those from the beginner tutorials at peterscarfe.com (2). They have been adapted and changed to demonstrate the usage and possibilities available with the VSHD.

# 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. The hardware display setup will therefore depend on which script you wish to run. Different setup options are described in the accompanying document "Display Setup for VisualSystem HD" (3). The 2D example requires monoscopic view, so setup 1M or 2M is required. In contrast, the fake and real 3D scripts require stereoscopic view with combined displays. This means that setup 2S Combined must be used.

# 3 Installation for Widows

While Psychtoolbox and the scripts can be run using either Matlab or Octave, this guide uses Octave by way of demonstration.

- GNU Octave: Download (gnu.org)
- Psychtoolbox: Psychtoolbox-3 - Download, Installation, and Update

Make sure you follow the installation instructions carefully, especially for Psychtoolbox. It is advisable to read through the instructions thoroughly before starting.

The scripts can be found on GitHub (4). Clone/download the repository to get the scripts on your computer.

# 4 Display distortion correction

Due to the optical system in the VSHD Goggles, the display will appear distorted to the viewer. More specifically, the image reaching the eyes will exhibit a pincushion artifact (Figure 1a). This can be compensated for by applying a barrel (Figure 1b) pre-distortion to the images sent to the VSHD.

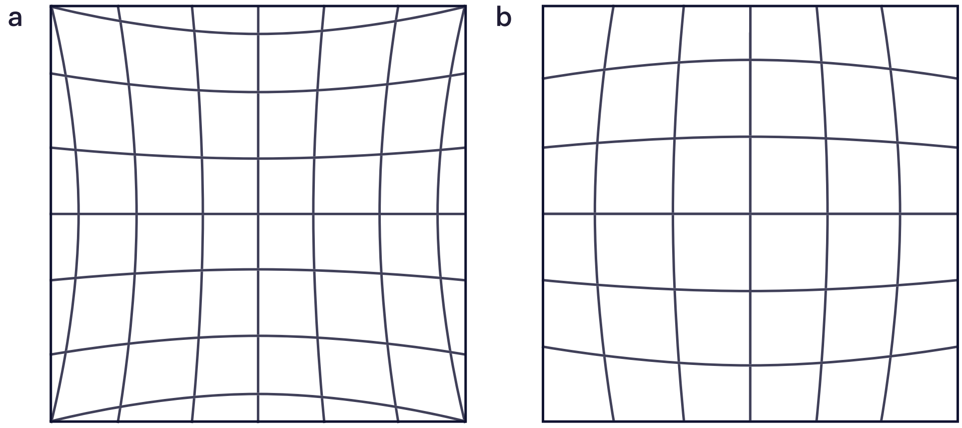


Figure 1: (a) Pincushion and (b) barrel distortion (5).

Figure 2 shows how an undistorted and pre-distorted image will look after passing through the VSHD Goggles' lenses. Adding pre-distortion will make the perceived image look undistorted.

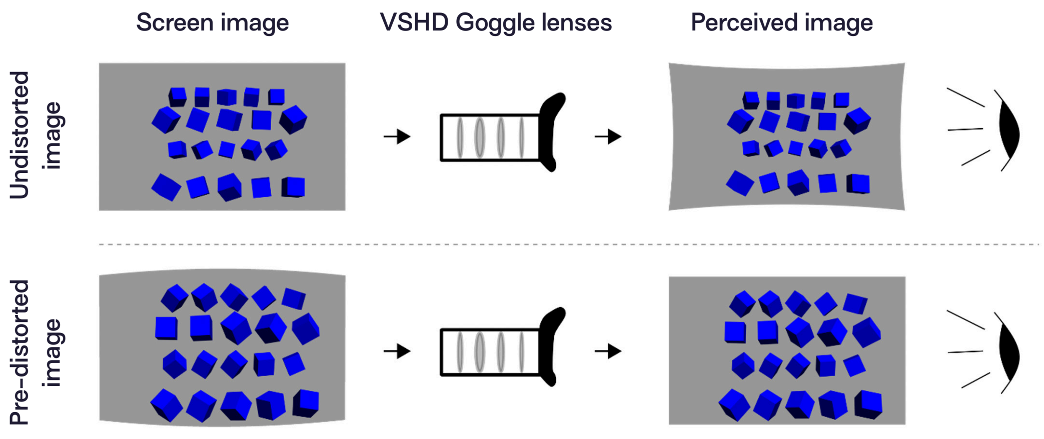


Figure 2: Perceived image with and without pre-distortion. The upper figure shows how an undistorted image will be distorted by the optical system. The lower figure shows how this is corrected for by adding barrel pre-distortion.

Psychtoolbox offers scripts which generate files that can be used to correct for the distortion by applying a pre-distortion. These work by displaying a grid of points which can be modified interactively until the grid is perceived to be undistorted. The corrections are then automatically saved to a file which can be loaded when setting up the screen in Octave/Matlab. Both the following section and the example scripts in section 5 show how to load the corrections. As the distortion will be approximately the same for every VSHD system, you can find a pre-made correction file (for diopter setting 0) in the GitHub repository with the example scripts (4).

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. To indicate how much diopter settings affect the distortion, the Psychtoolbox calibration was performed with three different diopter settings: -8, 0 and +5. The resulting barrel distortions for the three different settings are compared in Figure 3.

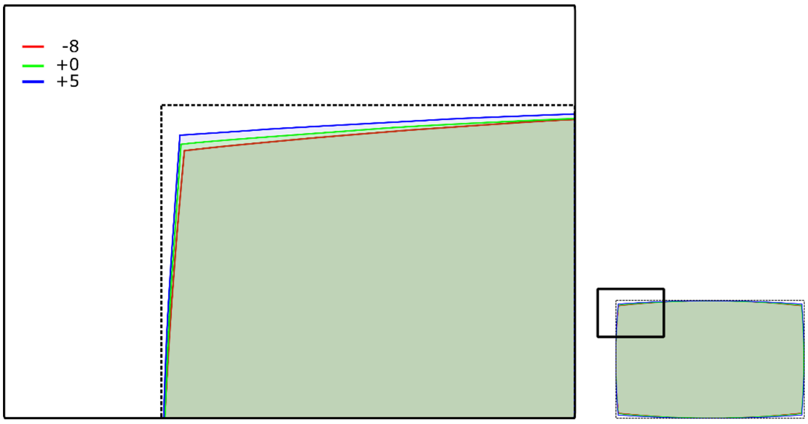


Figure 3: Barrel distortion for three different diopter settings on the VSHD.
The dashed line corresponds to the size of the full display.

As seen in Figure 3, there are noticeable differences when using the three different diopter settings.

Distortion can be quantified in a number of ways, for example by measuring the distance between points on an image versus their theoretical position (geometric distortion) and/or by measuring the difference in height between the corners and the center of a square (TV distortion (6)). To numerically compare the barrel distortion effects, geometric and TV distortions were measured. The results are shown in Table 1.

| Diopter Setting | TV Distortion | Geometric distortion |
|---|---|---|
| -8 | 4.15% | -4.29% |
| 0 | 3.47% | -3.63% |
| 5 | 2.71% | -3.22% |

Table 1: Distortion for different diopter settings.

We see from Table 1 that the difference in distortion between diopter settings of -8 and +5 is less than 1.5%. If a default correction file with a diopter setting of 0 is used, the distortion will be less than 0.8% at the extreme diopter values of -8 and +5. The user should evaluate if this is significant enough to warrant separate correction files for different diopter settings.

## 4.1 Making your own correction file

Psychtoolbox recommends using the function DisplayUndistortionBVL (7) to correct for distortion. This will display a grid of points which you can adjust by applying different transformations. The goal is to make the grid as large and rectilinear as possible. Once this is achieved, it will automatically save a file which can be loaded when setting up the display for the goggles.

The first step is to open the folder in which you want the correction file to be saved. A convenient solution is to choose the same location as your Matlab scripts. Once the program is open, click the blue folder button and choose the Current Directory as illustrated in Figure 4.

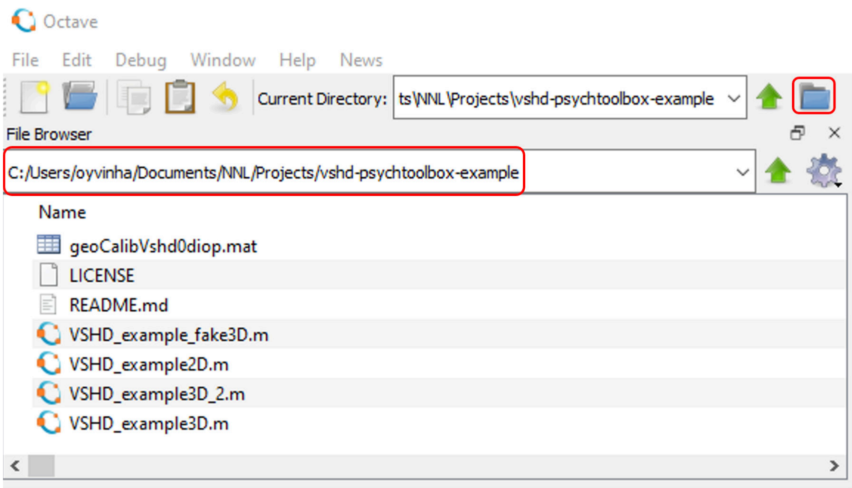

Figure 4: Choosing the folder for the correction file for GNU Octave.

Once the Current Directory has been set the calibration can be run. Start by opening the Command Window in Octave. This is shown in Figure 5.

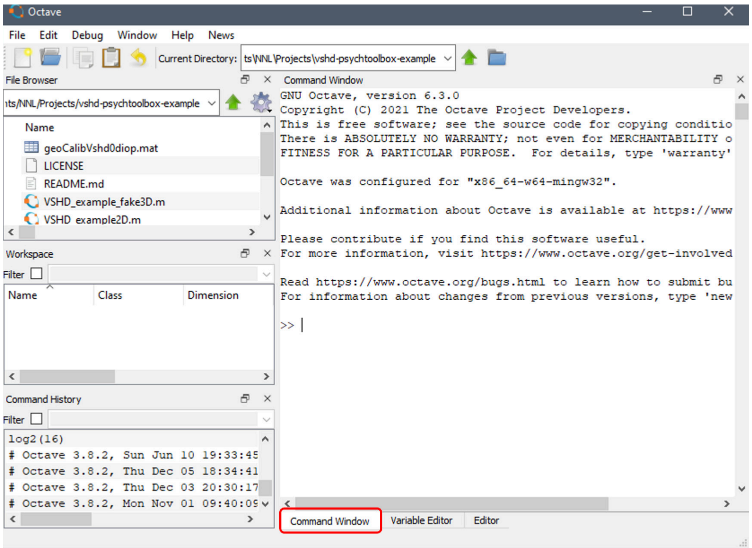

Figure 5: Command window in GNU Octave.

Run the “DisplayUndistortionBVL” script by typing the following into the command window. Note that the input parameters may differ depending on the setup, so read the description below before entering the command.

```
>> DisplayUndistortionBVL('calibData.mat', 1, [], [], [], 4);
```

As can be seen in the line above, the script takes in six optional inputs, three of which are used here. The first is the output calibration file name. Here, “calibData.mat” is used, but any name can be used. Remember to have the “.mat” extension at the end. The second parameter is the number of the display the calibration should be run on. This may be between one and the number of monitors available. Choose the one which corresponds to the VSHD display you wish to calibrate. Note that Psychtoolbox only allows the use of a single display ID, but this is not a problem since the same transformation will be applied to both eyes. The sixth and last parameter sets the stereo mode. This parameter depends on which setup you are using for the display. For stereoscopic view on the VSHD Goggles, this must be set to 4. For monoscopic view, this may be left empty “[]” or set to 0. Note that the resulting calibration file can be used for both mono- and stereoscopic setup, meaning a file made for stereoscopic setup can be used for monoscopic and vice versa. This is because the same transformation is applied to both eyes.

Upon entering the command, the calibration script will run. If the entered filename already exists, you will be asked in the Command Window to confirm whether you want to overwrite it. If a file of that name does not exist, or you choose to overwrite it, the calibration screen in Figure 6 will appear.

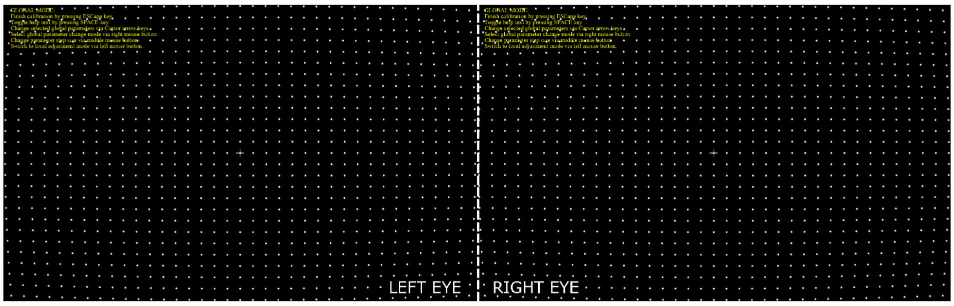


Figure 6: The calibration screen that should be displayed on the VSHD Goggles. Here the VSHD is set up as a combined display, and the left half of the image is displayed on the left eye and vice versa.

As mentioned, make sure that the screen ID is correct so that the calibration screen will show on the VSHD Goggles. Once this is achieved, follow the calibration instructions described in the “DisplayUndistortionBVL” documentation (7). A general tip is to start by increasing the step size for parameter adjustments by pressing the “middle mouse” button or the “m” key. By starting with a larger increment, it is easier to see how each parameter changes the grid. Adjusting the scaling and barrel/pincushion distortion will have the greatest effect on the distortion but try adjusting the other settings as well to become familiar with the procedure. The goal is, as mentioned previously, to make the grid as rectilinear and large as possible.

After all adjustments have been made, press “Esc” to finish. The script will then save the correction file automatically to the previously chosen folder. This file can then be loaded in the stimulus presentation scripts by adding the following lines before opening the window:

```
% Correct for distortion with calibration file
calibFilename = 'calibData.mat';
PsychImaging('PrepareConfiguration');
PsychImaging('AddTask', 'AllViews', 'GeometryCorrection', calibFilename);
```

The scripts provided show these commands. In Figure 7 we see the image before being displayed on the VSHD without (left) and with (right) the correction file loaded.

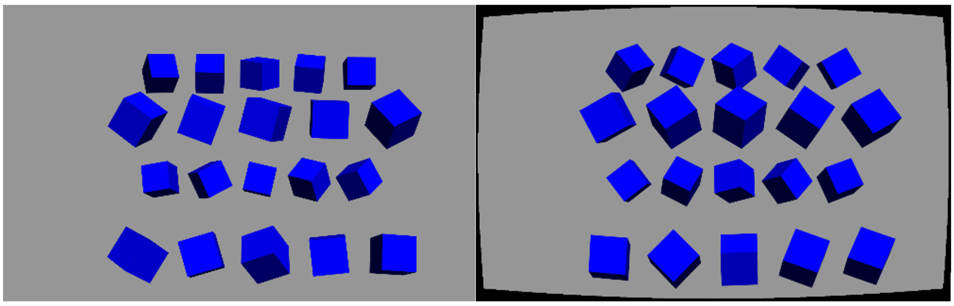

Figure 7: Example image with and without pre-distortion, before being sent to the VSHD. In the left image the correction file has not been loaded, so there is no pre-distortion. As shown in Figure 2, this image will appear pincushion distorted for the viewer. In the right image the correction file has been loaded. The barrel distortion compensates for the pincushion distortion introduced by the VSHD Goggles. The resulting perceived image will be undistorted.

By loading the correction file in the scripts, the applied pre-distortion will make the perceived image look undistorted. The applied barrel distortion is cancelled out by the pincushion distortion introduced by the optical system.

The applied pre-distortion does come at a cost, however. As the image is “squeezed” by the barrel distortion and then expanded out again by the optical system, the very edges of the image will exhibit a slight blurring. This results in a slightly lower visual fidelity.

# 5 Psychtoolbox examples

The following examples are based on the demos from peterscarfe.com (2), which are highly recommended to anyone wanting to learn basic Psychtoolbox programming. The scripts have been adapted to display the stimuli on the VSHD Goggles. There are four scripts, each showing different types of stimuli. The first shows 2D animations and text. The second shows how a fake 3D image can give depth perception without rendering 3D graphics. This is achieved using a slight spatial mismatch between the images for each eye, giving a sense of depth. Finally, the last two examples show how to display real 3D on the VSHD Goggles. In a real 3D scenario, the images for each eye are rendered from two different camera positions in the 3D scene.

## 5.1 2D example

The 2D example is based on the "Internal External Texture Rotation Demo" (2), and this has been adapted for use with the VSHD. This demo is designed for monoscopic view. To run it with the VSHD, you must therefore use either setup 1M or 2M (3).

The script shows rotating squares with texture, as shown in Figure 8.

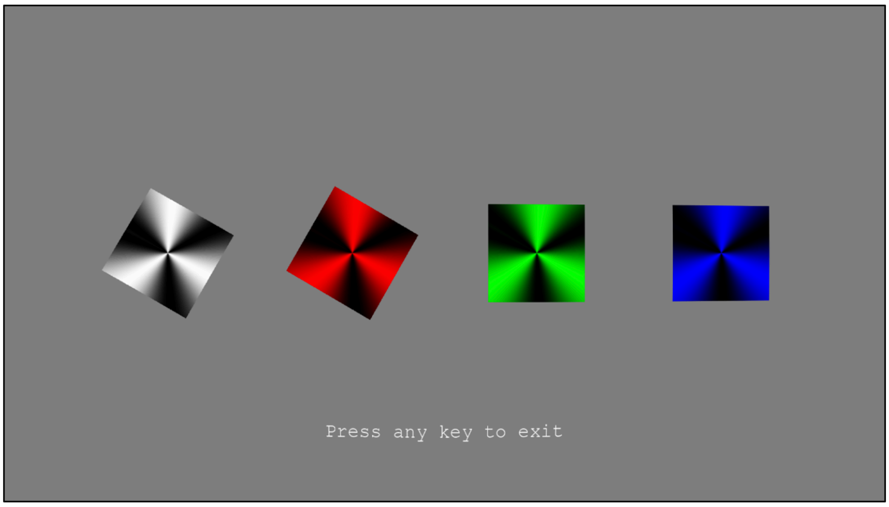


Figure 8: 2D stimuli example showing rotating cubes and textures.

The only changes made to the "Internal External Texture Rotation Demo" were the addition of distortion correction and some explanatory text. It has also been reorganized for structure and readability.

## 5.2 Fake 3D

This example is based on the principles of depth covered by the "Square In Depth Demo", but they have been applied to shapes created using other available demos (2). To achieve the depth perception, the script utilizes stereoscopic view. As Psychtoolbox can only display an image to one display ID, setup 2S Combined must be used for the VSHD (3).

The demo shows four different objects, as shown in Figure 9.

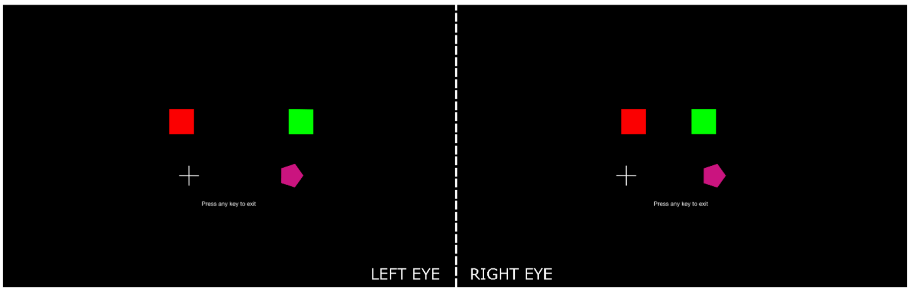


Figure 9: Fake 3D example. The left half of the image is sent to the left eye, and the right half to the right eye.

The images shown to the left and right eyes are different. To achieve the fake 3D appearance, slight horizontal shifts are applied to the different objects for the right eye compared with the left eye. This will make it appear as though the objects have depth. The amount of shift applied determines the extent of the perceived depth. It can be adjusted using the "shifterPix" variable in the script. A larger shift will make an object appear closer. Note that a too large shift will then make it difficult to focus on the object.

In this example a different shift has been applied to each of the objects, to make their depths appear different. This becomes very apparent when considering the two squares. When looking with only one eye, it is easy to see that they are both equal in size. When looking at the image with both eyes however, the green square appears smaller and closer. The major limitation of this 3D effect is that it only works for 2D non-overlapping objects, but the advantage is that you do not need to render 3D graphics.

## 5.3 Real 3D example 1

This VSHD example is based on the "Rotating Cubes Demo" (2). As with the fake 3D demo, it uses stereoscopic view, therefore the 2S Combined setup must be used for VSHD (3).

The demo shows rotating cubes moving back and forth, thus illustrating the 3D effect. A screenshot is shown in Figure 10.

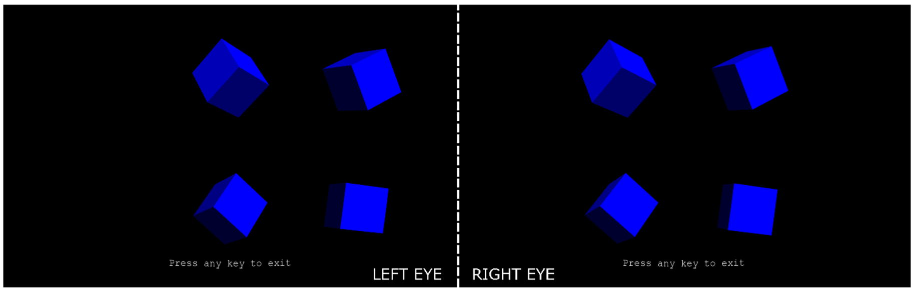


Figure 10: 3D example script showing rotating cubes moving back and forth.

As with the other scripts, informative text and distortion correction have been added.

Rendering of the 3D graphics is done using OpenGL and inbuilt functions in Psychtoolbox. To achieve the "real" 3D effect, two different images are rendered for each eye. The original script was written for a monoscopic view, but has been adapted to use stereoscopic display in this case. To render different images for each eye, two different camera positions with a horizontal offset between them are used. This offset imitates the offset between the human eyes. The images will be slightly shifted, and the objects seen from slightly different angles, thus the objects will be perceived to be in 3D space.

To ensure proper scaling and to achieve a proper view of the objects, the field of view (FOV) and inter pupil distance (IPD) must be set to appropriate values. The FOV in the script is set to that of the VSHD Goggles (34° vertical) for a diopter setting of 0. Distances in OpenGL are unitless, thus for simplicity, all values are assumed to be in centimeters. The IPD is set to 6.3 cm in the script but should be adjusted to the actual IPD of the subject by changing the IPD variable. Note that the sizes of the objects and distances between them in the scene are then also assumed to be in centimeters to ensure the correct scaling.

## 5.4 Real 3D example 2

This example is based on the previous 3D example but has been extended to display different objects in different colors. Instead of the objects moving back and forth, the camera is now moving side to side to illustrate the 3D effect. As with the previous 3D example, this script requires setup 2S Combined for VSHD (3). A screenshot of the demo is shown in Figure 11.

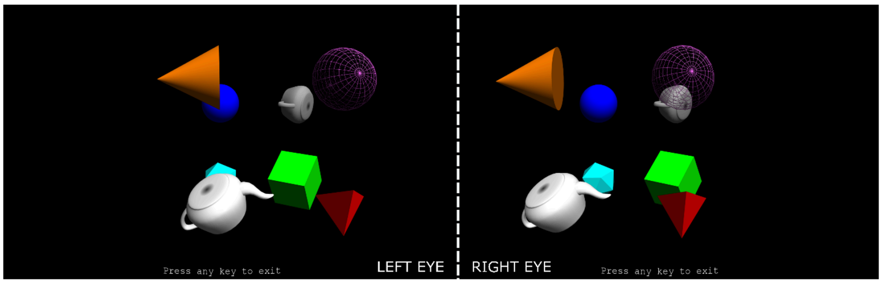


Figure 11: 3D example showing different shapes in different colors.

The different shapes are drawn using geometric object rendering functions from OpenGL Utility Toolbox (8). Objects can be changed easily in the script. As with the earlier 3D demo, FOV and IPD must be set correctly to achieve the most realistic experience.

# 6 References


1. Psychtoolbox. Overview. psychtoolbox.org

   [Online] June 10, 2021. [Cited: July 28, 2021.] http://psychtoolbox.org/overview.html

2. Scarfe, Peter. Psychtoolbox Demos. peterscarfe.com

   [Online] [Cited: July 28, 2021.] http://peterscarfe.com/ptbtutorials.html

3. Display Setup for VisualSystem HD. NordicNeuroLab

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

4. —. vshd-psychtoolbox-example. GitHub

   [Online] https://github.com/nordicneurolab/vshd-psychtoolbox-example

5. Distortion. wikipedia.com

   [Online] July 1, 2021. [Cited: July 28, 2021.] https://en.wikipedia.org/wiki/Distortion_(optics)

6. Distrotion. edmundoptics.com

   [Online] March 14, 2014. https://www.edmundoptics.com/knowledge-center/application-notes/imaging/distortion/

7. DisplayUndistortionBVL. psychtoolbox.org

   [Online] June 10, 2021. [Cited: July 28, 2021.] http://psychtoolbox.org/docs/DisplayUndistortionBVL

8. Kilgard, Mark. Geometric Object Rendering. opengl.org

   [Online] February 23, 1996. [Cited: July 28, 2021.] https://www.opengl.org/resources/libraries/glut/spec3/node80.html

Nordic Neuro Lab

CE

VisualSystem HD

Headquarters
NordicNeuroLab

Møllendalsveien 1
N-5009 Bergen
Norway

Regional Office
NordicNeuroLab, Inc.

234 W. Florida St., Suite 205
Milwaukee, WI 53204
USA

NL-100-0091-01