THE UNIVERSITY OF ARIZONA
HEALTH SCIENCES
Sensor Lab

ZED 2/2i Quick Start

Note: The ZED 2 and the ZED 2i are both available at the UA Sensor Lab. The ZED2i is a newer version with an improved lens, polarizing filter, and an IP66-rated enclosure (making it dust and humidity resistant). **This quick start page supports both ZED 2 and ZED 2i cameras.**

Intro to the Capabilities of ZED Cameras
ZED cameras are passive depth sensors, meaning they use no lasers or IR light (which have limited range and suffer sunlight interference). Depth range of the ZED2 and ZED2i is **20m in 3D** and up to **40m in 2D**. The ZED SDK gives you capabilities such as object detection, skeleton tracking, spatial mapping, positional tracking for SLAM(Simultaneous localization and mapping).

This Quick Start guide was written in **March 2023**. If any of the links or information become outdated, please contact the UA Sensor Lab.

Getting Started

1. **Download the ZED SDK** here.
   a. You may also be prompted to download Nvidia Cuda and/or Visual Studio Code (VS Code). This download process may take 10+ minutes.
2. **Plug in your ZED camera** using the USB cable provided. If you have the ZED 2, screw the mini tripod into the bottom of the camera.
   a. To minimize delay between when data is captured and saved, plug the ZED camera into a USB 3.0 receptor, which may be on the back of your computer.
3. **Download starter code** and follow introductory tutorials in C, C++, C#, or Python here.
   a. If you use a Python tutorial and get the error `ModuleNotFoundError: No module named 'pyzed'` when you try to run the code, get help here.
   b. For help on specific Python tutorials and saving images/data, go here.
4. Explore the ZED Explorer, ZED Sensor Viewer, ZEDfu, and ZED Depth Viewer you installed in step 1.
5. See the ZED API Documentation for more information.

Help with Python pyzed.s1

If you try running Python starter code and get the error `ModuleNotFoundError: No module named 'pyzed',` you'll need to install Anaconda and open the starter code in a Python environment with built-in libraries.

1. Download Anaconda here.
2. Download ZED_Env_Anaconda.yaml from the UA Sensor Lab page. The contents of this file are also available here.
3. Open Anaconda Navigator and select **Environments** from the left bar. Click **Import** at the bottom of the screen, and select the YAML file downloaded in step 2.
4. Click the green play button to the right of ZED_Env_Anaconda and select **Open Terminal**.
5. In the terminal, copy the following command and hit enter:

```
python -m pip install cython numpy opencv-python pyopengl
```

Continue with steps 6-8 below, or follow the same instructions with more visuals here.

6. Navigate to the folder on your computer where ZED SDK was downloaded (by default, it will be under Program Files (x86) > ZED SDK). Copy and paste the get_python_api.py file from this ZED SDK folder into your Downloads folder.
7. Open terminal in your Downloads and run get_python_api.py
   a. Open Terminal/Command Prompt and type the following command and hit enter: `cd Downloads`
   b. Copy and paste the following command and hit enter:

   ```
   python get_python_api.py
   ```

   Under Anaconda Navigator > Environments > ZED_Env_Anaconda, you should see `cython, numpy, opencv-python, pyopengl, pyopengl-accelerate`, and `pyzed` installed. You may need to navigate back to the base (root) environment and then click on the ZED_Env_Anaconda environment to see all of these packages.
8. In the Anaconda Navigator, go to Home and ensure that ZED_Env_Anaconda is selected on the top bar. Find the VS Code box and click Launch. In VS Code, open and run your Python file.

You can now continue on to step 4 under Getting Started.

Tutorial Help

The ZED tutorials found here are a great place to start when setting up ZED cameras, but they don't show you how to save data. The following tutorials supplement the ZED tutorials and can help you extract images and CSV files from your camera.

Image Capture: This ZED tutorial captures 50 images but doesn't save them. The following steps will allow you to capture JPG images or numpy arrays and save them with timestamps in their names.

1. Add the following imports outside of main():

```
import time
import numpy as np
```

2. Change the while-loop inside of main() to resemble the following code.

```
    # Capture 50 frames and stop
    i = 0
    image = sl.Mat()
    runtime_parameters = sl.RuntimeParameters()
    while i < 50:
        if zed.grab(runtime_parameters) == sl.ERROR_CODE.SUCCESS:
            # A new image is available if grab() returns SUCCESS
            zed.retrieve_image(image, sl.VIEW.LEFT)

            # Save images as JPG
            img_name = 'images/myProject_{}.jpg'.format(time.time())
            image.write(img_name)

            # Save images as numpy arrays
            img = image.get_data()
            np.save(img_name, img)
```

3. Make a folder called 'images' in your working directory. When you run the code, 50 JPG images and 50 numpy files will be saved in your new 'images' folder.

Depth Perception: This ZED tutorial prints the distance of the object in the center of the camera's view 150 times, but it doesn't save any data. The following steps will allow you to save depth data into a CSV file.

1. Add the following import outside of main(): `import time`
2. Inside of main() and before the while-loop, use the following code to create a csv file. The f.write() line gives header names for the CSV file.

```
    # Create a file to save time, x, y, and distance
    f = open("yourFileName.csv", "a")
    f.write('img_index,time,x,y,distance\n')
```

Tip: Change the “a” parameter to “w” to overwrite the file’s data every time you run the code. The “a” causes new data to append to the CSV file every time.

3. Inside of main() and after the distance variable is created, write the data to your csv file.

```
# Write data to the csv file
line = str(i) + "," + str(time.time()) + "," + str(x) + "," + str(y) + "," +
str(distance) + "\n"
f.write(line)
```

4. Outside of the while-loop, close the file.

```
f.close()
```

Camera/Positional Tracking: This ZED tutorial prints translation, orientation, acceleration, and velocity data but doesn’t save it. The following steps will allow you to save this data to a CSV file.

1. Inside of main() and before the while-loop, use the following code to create a csv file. The f.write() line gives header names for the CSV file.

```
    # Create a file to save data
    f = open("positionalData.csv", "w")
    f.write('timestamp,Tx,Ty,Tz,Ox,Oy,Oz,Ow,Ax,Ay,Az,Vx,Vy,Vz\n')
```

2. Inside the while-loop, use the following code to save the data to the CSV file.

```
# Save data to csv file
data = str(zed_pose.timestamp.get_milliseconds()) + "," +\
       str(tx) + "," + str(ty) + "," + str(tz) + "," +\
       str(ox) + "," + str(oy) + "," + str(oz) + str(ow) + "," +\
       str(ax) + "," + str(ay) + "," + str(az) + "," +\
       str(vx) + "," + str(vy) + "," + str(vz) + "\n"
f.write(data)
```

3. Outside of the while-loop, close the file.

```
f.close()
```

Contents of ZED_Env_Anaconda.yaml

```
name: ZED_env
channels:
 - defaults
dependencies:
 - ca-certificates=2023.01.10=haa95532_0
 - certifi=2022.12.7=py39haa95532_0
 - openssl=1.1.1t=h2bbff1b_0
 - pip=23.0.1=py39haa95532_0
 - python=3.9.16=h6244533_2
 - setuptools=65.6.3=py39haa95532_0
 - sqlite=3.41.1=h2bbff1b_0
 - tzdata=2022g=h04d1e81_0
 - vc=14.2=h21ff451_1
 - vs2015_runtime=14.27.29016=h5e58377_2
 - wheel=0.38.4=py39haa95532_0
 - wincertstore=0.2=py39haa95532_2
 - pip:
   - cython==0.29.33
   - numpy==1.24.2
   - opencv-python==4.7.0.72
   - pyopengl==3.1.5
   - pyopengl-accelerate==3.1.5
   - pyzed==3.8
```