STANDARD OPERATING PROCEDURE (SOP) For Operating Zed2I Camera

Title: Complete Setup and Operational Guidelines for ZED 2i Camera

Prepared by: Vedant Rakesh Mukhekar [Intern]

Faculty Mentor's : -

Kaushal Sharma[Scientist E & Chief Investigator UAS/ Drone Project]

Harshada Deshpande [Scientist B & CO-Chief Investigator UAS/ Drone Project]

Aaditya Khapre [Project Engineer UAS/ Drone Project]

Aman Shrivastava [Project Engineer UAS/ Drone Project]

Mihir Harne [Project Engineer UAS/ Drone Project]

Date: 6th October 2025.

---

1. Objective

The objective of this Standard Operating Procedure (SOP) is to provide a **comprehensive guide** for the installation, configuration, and operation of the **Stereolabs ZED 2i Stereo Camera**.
This document ensures that **any user, regardless of prior experience**, can successfully install the required drivers, SDKs, and software, and begin utilizing the ZED 2i for various computer vision and AI applications, including depth estimation, spatial mapping, object detection, and 3D reconstruction.

---

2. Scope

This SOP is applicable to:

- AI and robotics researchers, developers, and students.
- Engineers integrating the ZED 2i into computer vision systems.
- Laboratories or institutions deploying ZED-based depth perception, mapping, or navigation systems.
- Developers implementing ZED 2i on both **Windows** and **Ubuntu (Linux)** systems.

This SOP covers:

1. Hardware setup and connectivity.
2. CUDA and ZED SDK installation.
3. Python environment setup and dependencies.
4. Functional testing using ZED Explorer and ZEDfu.
5. Python-based camera operations.
6. Overview of additional ZED applications and use cases.

---

3. Equipment and System Requirements

3.1 Hardware

- **ZED 2i Stereo Camera**
- **Computer or workstation** with:
    - NVIDIA GPU supporting **CUDA 11.8 or higher**
    - Minimum 4 GB VRAM (8 GB or more recommended)
    - At least USB 3.0 port
- (Optional) **Tripod or mount** for stable operation

3.2 Software Requirements

Component	Recommended Version	Source
Operating System	Windows 10/11 (64-bit) / Ubuntu 20.04+	Microsoft / Ubuntu
NVIDIA CUDA Toolkit	11.8	https://developer.nvidia.com/cuda-toolkit
ZED SDK	Version matching CUDA 11.8	https://www.stereolabs.com/developers
Python	3.8 – 3.10 (64-bit)	https://www.python.org/downloads
Visual Studio Build Tools (Windows)	2019 or 2022	Microsoft

4. Pre-Installation Preparation

1. **Check GPU compatibility:**

Open Command Prompt and run:

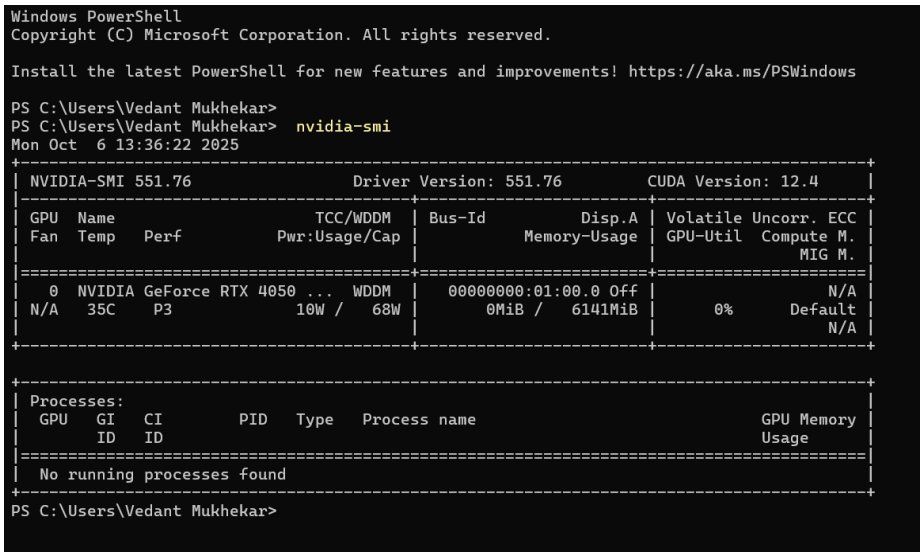
```
nvidia-smi
```

   - Verify the GPU is detected and the driver version supports CUDA 11.8. as seen in Fig 1.

2. **Uninstall any old ZED SDK versions** to prevent conflicts.

3. **Ensure stable power supply and internet connection** throughout installation.

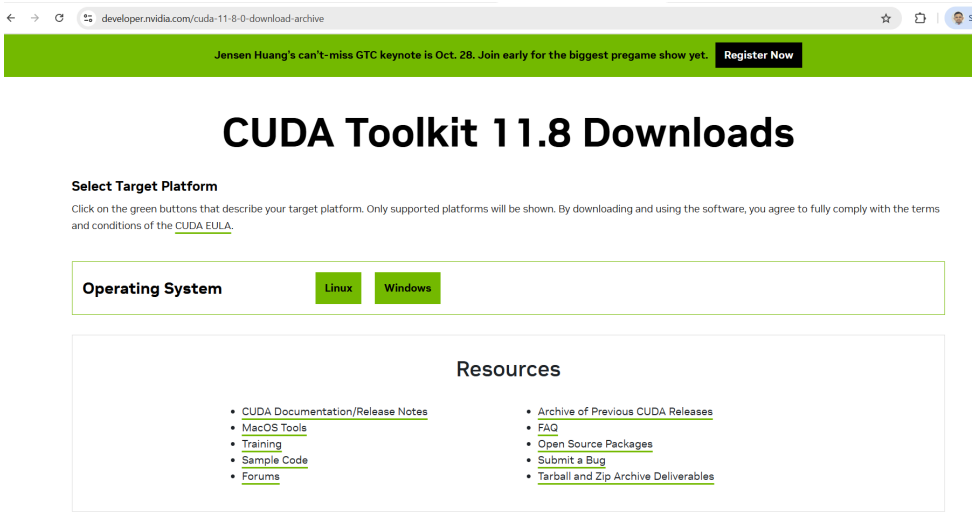
4. **Create a dedicated installation directory** (e.g., `C:\ZED_SETUP` or `~/zed_setup`).



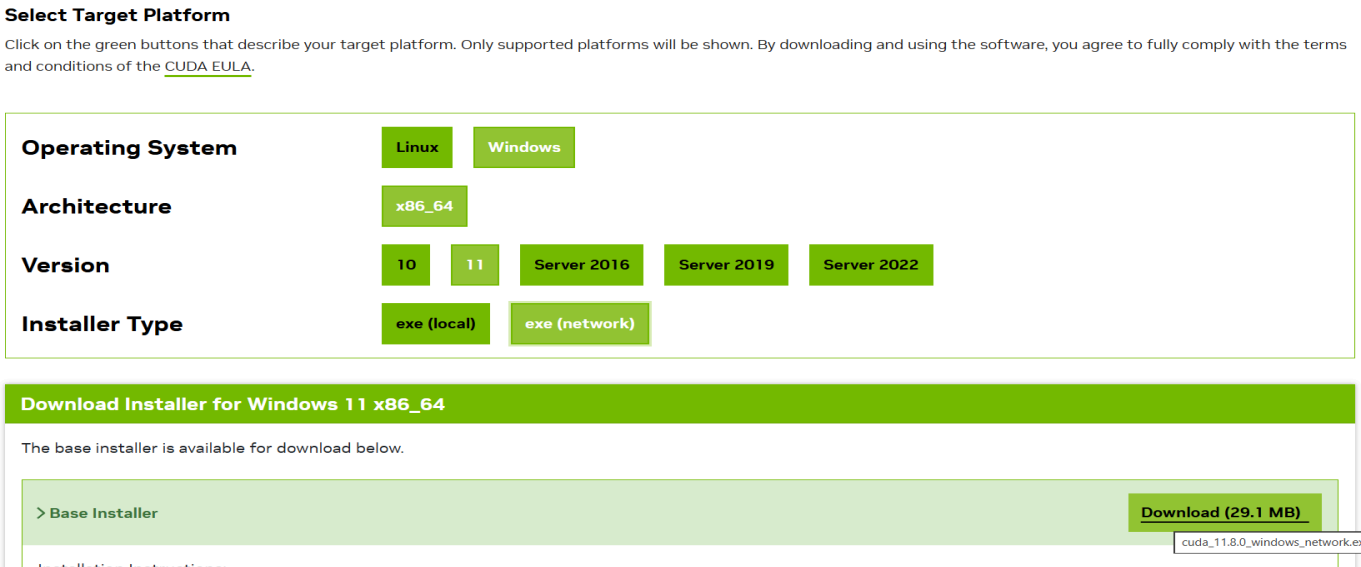
5. Installation Procedure

5.1 Installing CUDA Toolkit 11.8

1. Navigate to NVIDIA CUDA Toolkit Downloads 11.8



2. Select:
    - Operating System: Windows / Linux
    - Architecture: x86_64
    - Version: 11.8
    - Installer Type: `exe (network)` for Windows or `.run` for Ubuntu.

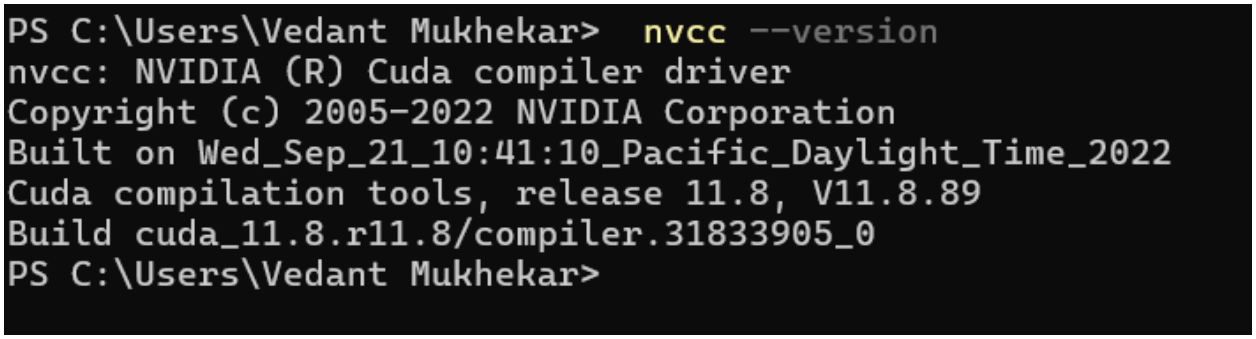


3. Run the installer.
    - Choose **Express Installation**.
    - Follow on-screen prompts.
    - Restart the system once installation completes.

4. Verify installation:

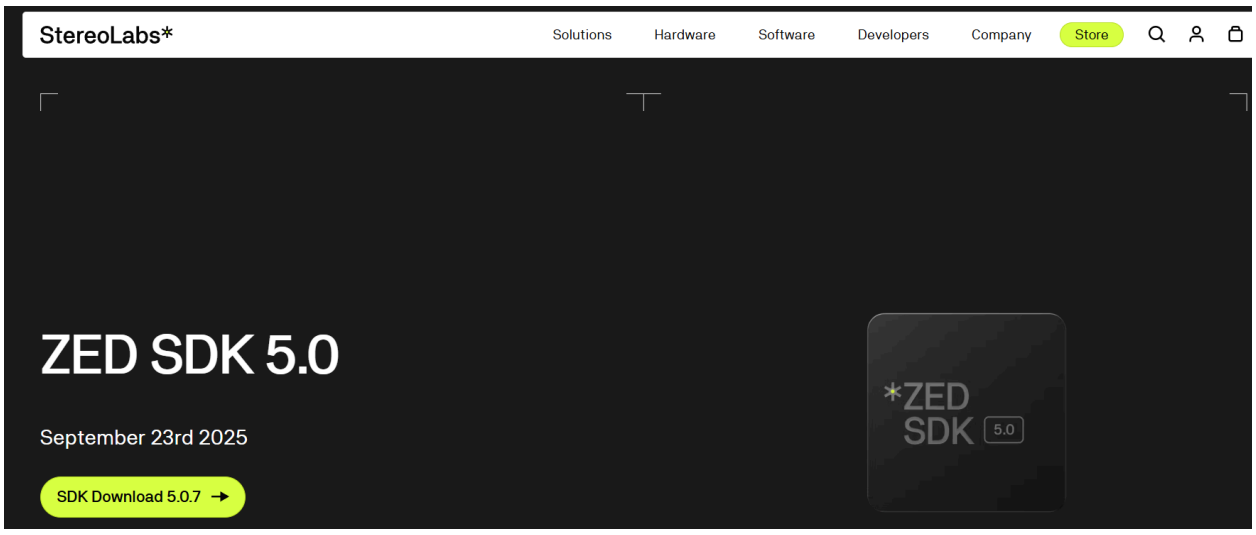
```
nvcc --version
```

The version output should display CUDA 11.8.

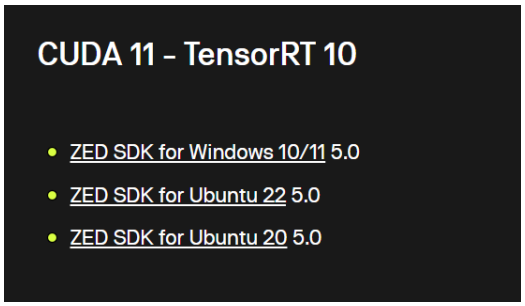


5.2 Installing ZED SDK

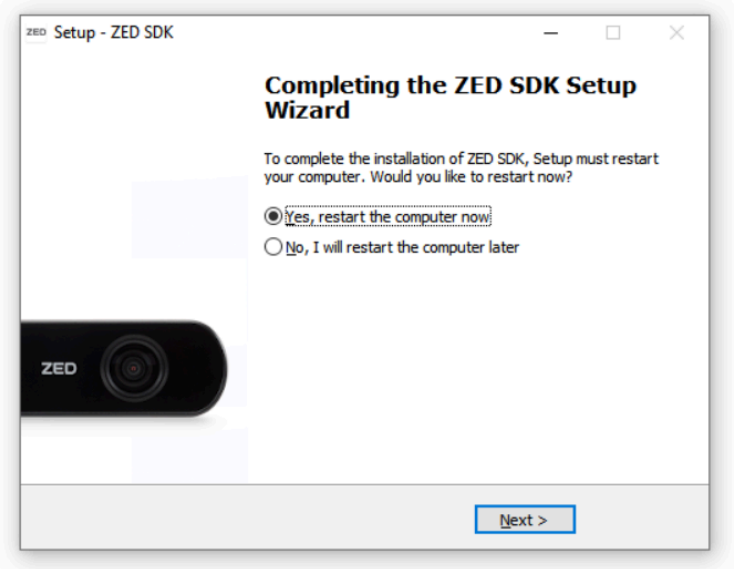
1. Visit Stereolabs ZED SDK Downloads.

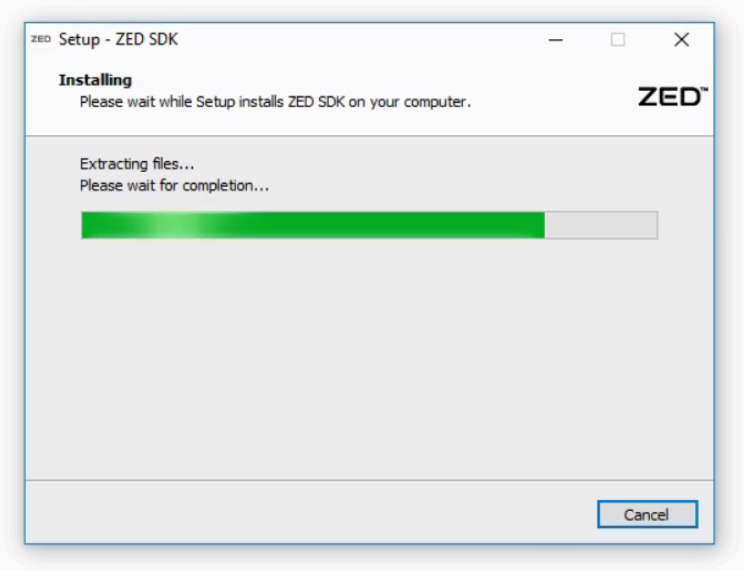


2. Choose the SDK version compatible with **CUDA 11.8**.



3. Download the **Windows executable** (`.exe`) or **Ubuntu installer** (`.run.zstd`).
4. **Installation Steps (Windows):**
    - Run the `.exe` file.
    - Accept license terms.
    - Use default installation directory: `C:\Program Files (x86)\ZED SDK`.
    - Allow the installer to add PATH variables automatically.
    - Reboot the system.





5. **Installation Steps (Ubuntu):**

Open terminal and navigate to download folder.

```
chmod +x ZED_SDK_Ubuntu20_cuda11.8_vX.Y.Z.zstd.run
sudo ./ZED_SDK_Ubuntu20_cuda11.8_vX.Y.Z.zstd.run
```

- Follow on-screen instructions.
- Reboot when prompted.

ZED SDK Applications and Tools

Once the SDK installation is complete, you can access all tools from:
Start Menu → Stereolabs → ZED SDK → Tools

Start Menu → ZED SDK → Tools

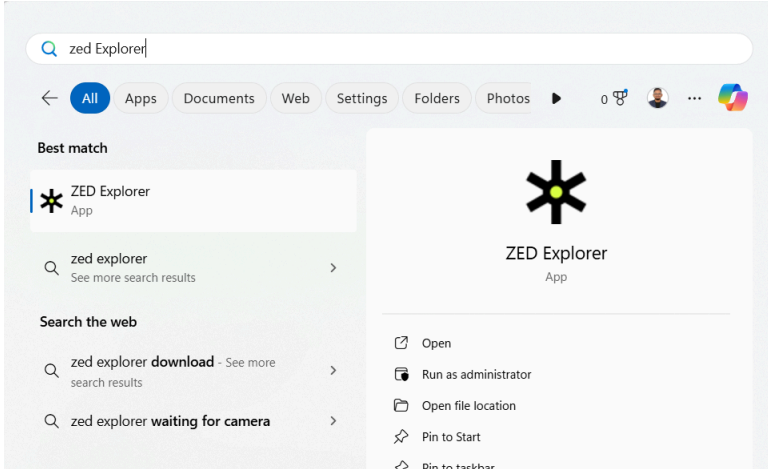
Below is a description of each major utility.

5.1 ZED Explorer

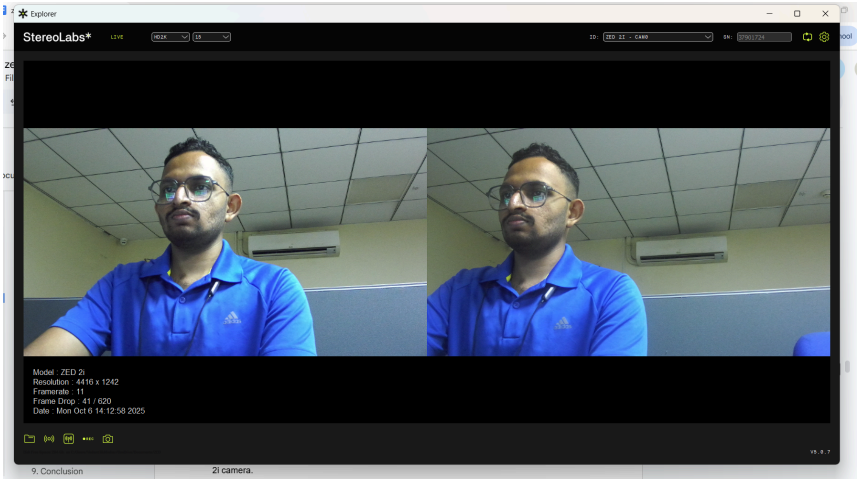
Purpose:
ZED Explorer is the main application used to view live stereo feed from the camera, adjust exposure, gain, resolution, and save recordings.

Steps:

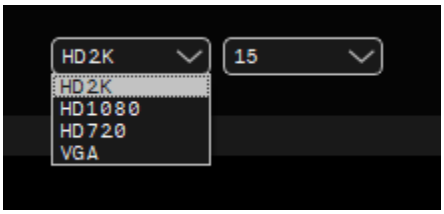
1. Open **ZED Explorer after searching it from the start Menu**.



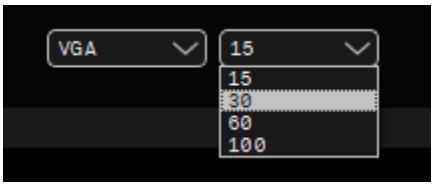
2. Connect the ZED 2i camera via USB 3.0 port(provided with zed2i camera).
3. The live stereo view appears — left and right camera frames side-by-side.



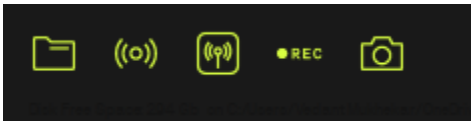
4. You can:
    - Adjust **Resolution** (HD720, HD1080, 2K)



    - Modify **Frame Rate**



    - Start/Stop **Recording** (`.svo` format)



5. Output files (`.svo`) are stored locally for further use in SDK-based applications.

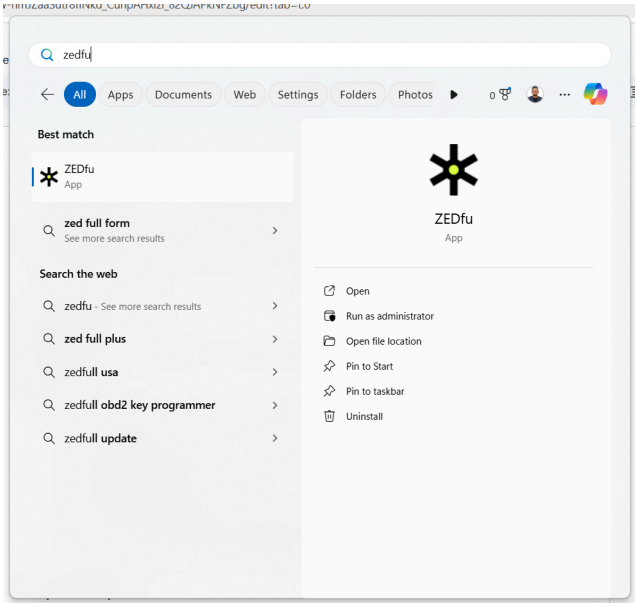
Expected Output:
A real-time dual-camera view with options to record and capture depth/pose data.

5.2 ZED-Fusion (ZEDfu)

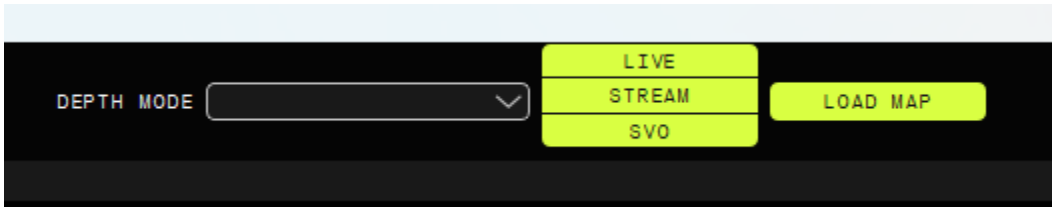
Purpose:
Used for 3D spatial mapping, point cloud generation, and scene reconstruction using the ZED 2i camera.

Steps:

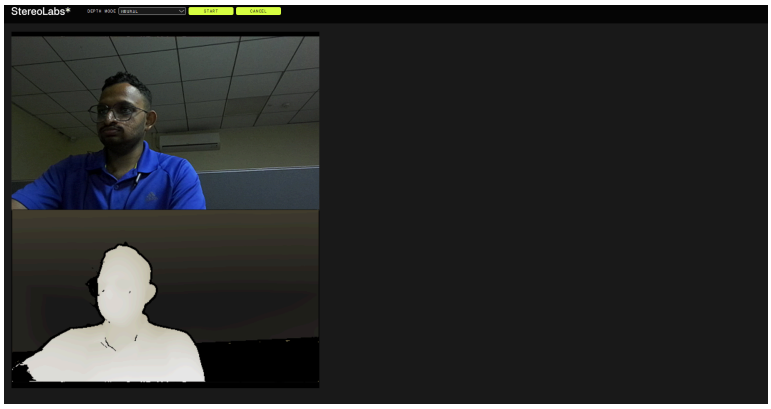
1. Launch **ZEDfu** from the ZED SDK tools **after searching it from the start Menu**.



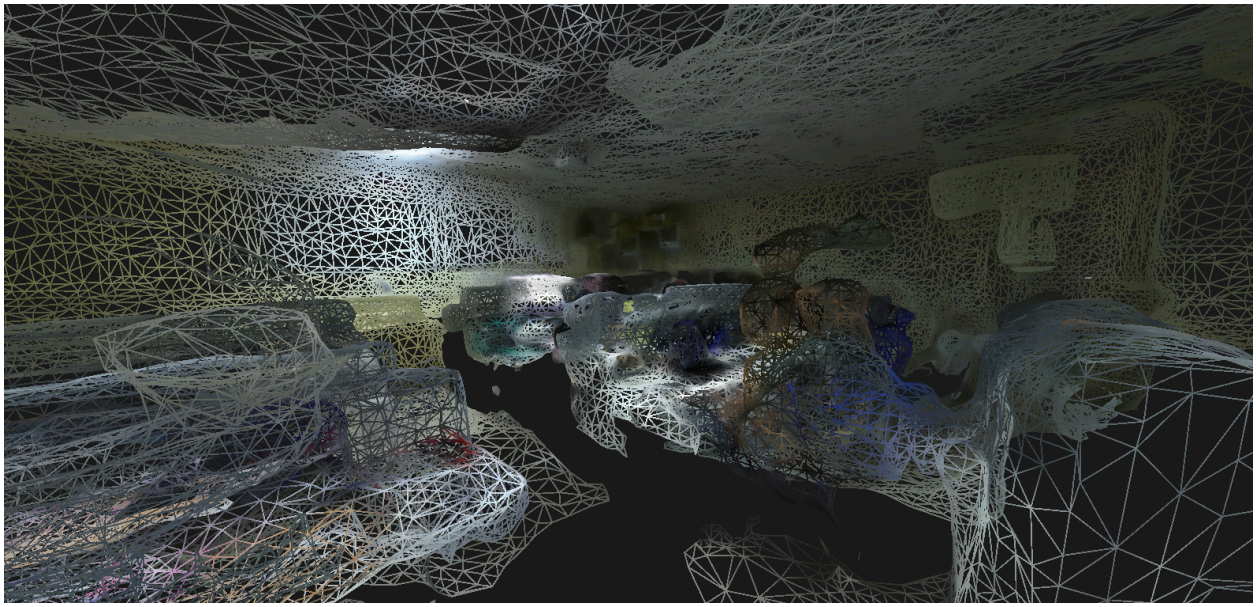
2. Select your ZED 2i device and resolution.



3. Click **Start** to begin scanning.



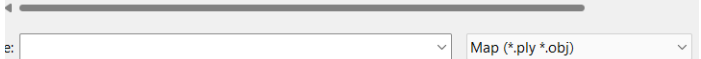
4. Move the camera slowly to map your environment.



5. Once done, click **Stop** and **Save Mesh/Point Cloud**.

Expected Output:

A 3D model reconstruction of the environment in `.obj` or `.ply` format.
You can rotate and visualize the 3D space interactively.

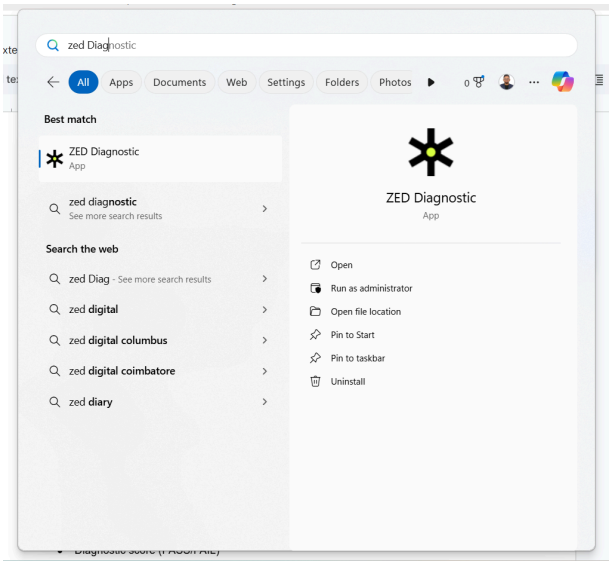


5.3 ZED Diagnostics

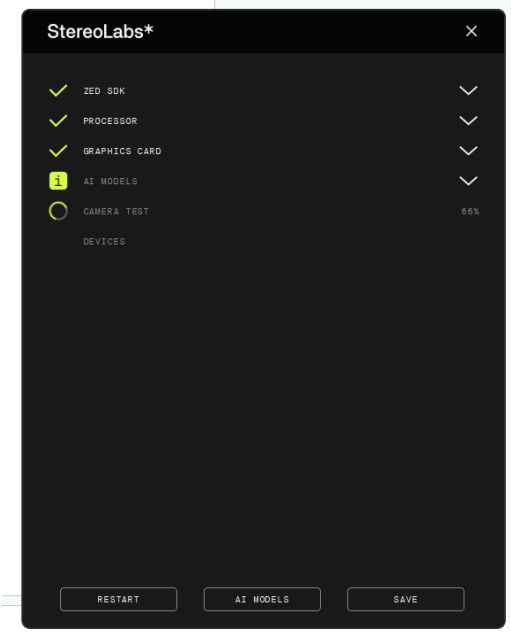
Purpose:
Checks hardware and software compatibility, driver versions, and sensor health.

Steps:

1. Open **ZED Diagnostics from Start Menu**.



2. Click **Run Diagnostics**.



3. Wait for system verification to complete.

Expected Output:
A detailed report showing:

- GPU details and driver version
- ZED SDK version
- USB connection type
- Camera firmware status
- Diagnostic score (PASS/FAIL)

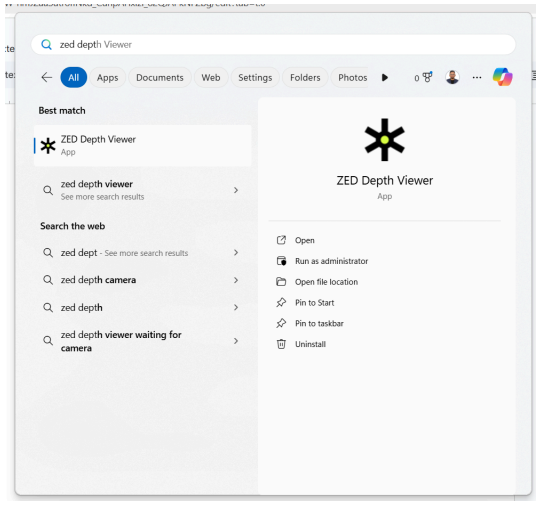
Use this before any serious deployment to ensure all systems are configured correctly.

5.4 ZED Depth Viewer

Purpose:
Displays real-time depth and disparity maps from the stereo camera.

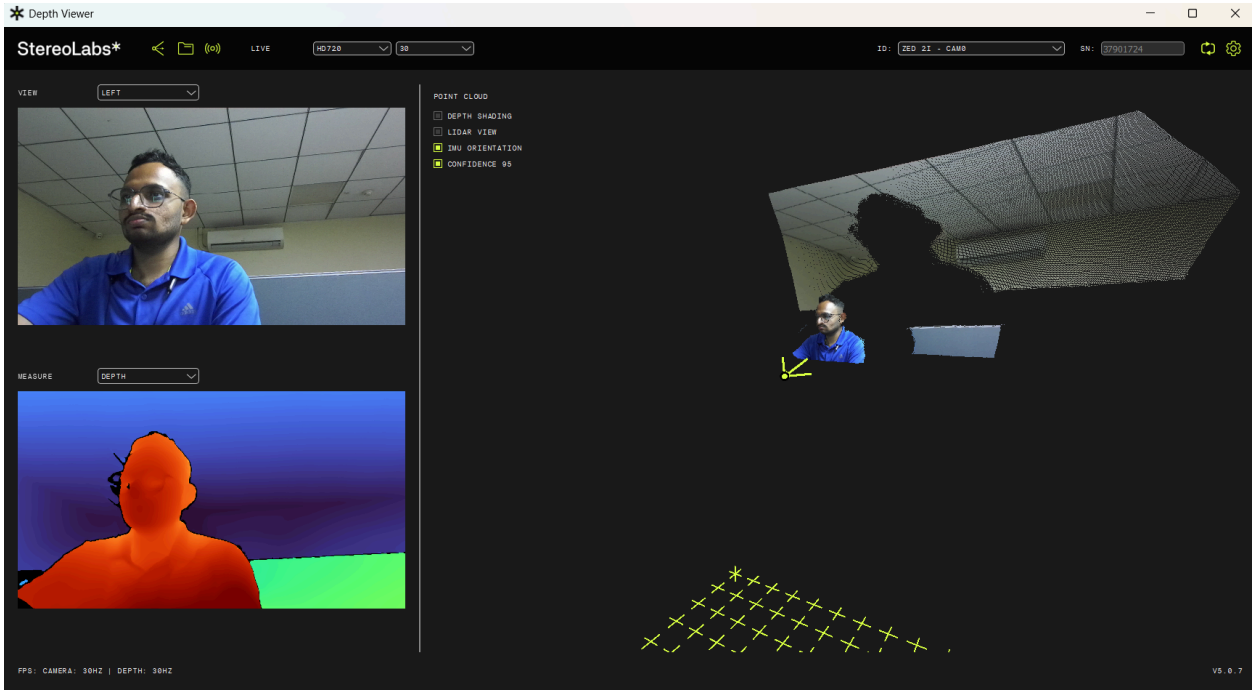
Steps:

1. Launch **ZED Depth Viewer from the Start Menu.**

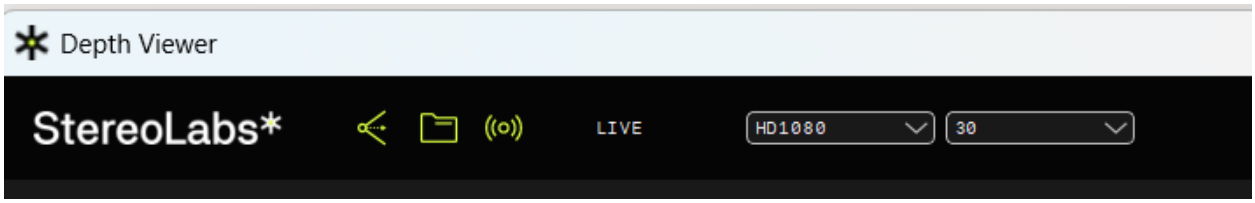


2. Choose **ZED 2i** and desired **Depth Mode** (Performance/Quality/Ultra).

3. Observe:
    - Left: RGB feed
    - Right: Depth visualization (colored by distance)



4. Optionally record or save frames.



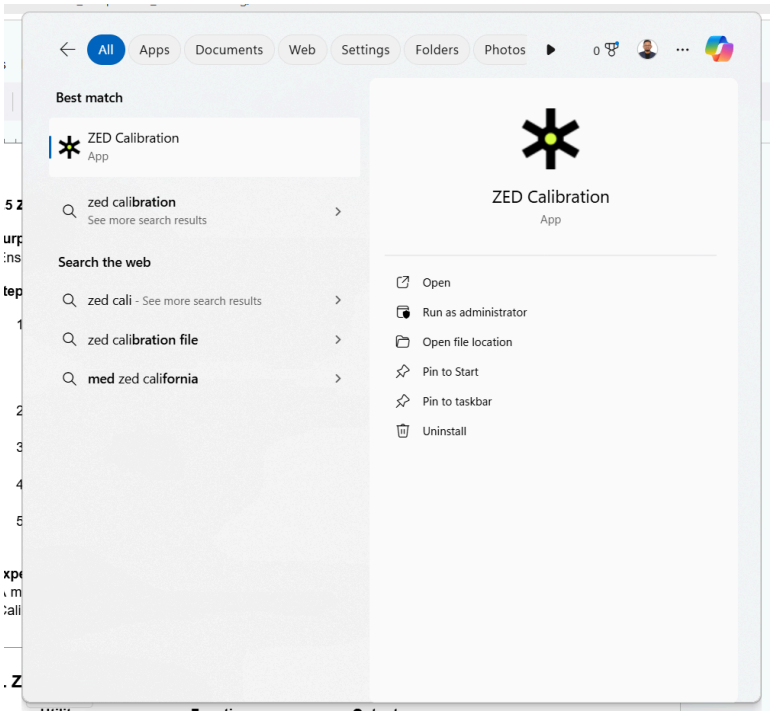
Expected Output:
Depth map showing color-coded distances — blue for far, red for near objects.
Used in perception and obstacle detection applications.

5.5 ZED Calibration Tool

Purpose:
Ensures stereo cameras are properly aligned and calibrated for accurate depth sensing.

Steps:

1. Open **ZED Calibration Tool from the Start Menu**.



2. Place a checkerboard calibration target in front of the camera.
3. Capture images as instructed.
4. Follow the on-screen calibration wizard.
5. Save the new calibration file.

Expected Output:
A message confirming calibration success and improved stereo alignment.
Calibration files are saved within the SDK's calibration folder.

---

6. ZED SDK Utilities Overview

Utility	Function	Output
ZED Explorer	Live camera view, recording	`.svo` files
ZEDfu	3D reconstruction	`.obj`, `.ply`
ZED Depth Viewer	Depth map visualization	Depth images
ZED Diagnostics	System verification	Diagnostic report
ZED Calibration	Stereo calibration	Calibration file

---

7. Example Python Code for ZED Camera Initialization

```
import pyzed.sl as sl

# Create a ZED camera object
zed = sl.Camera()

# Set configuration parameters
init_params = sl.InitParameters()
init_params.camera_resolution = sl.RESOLUTION.HD1080
init_params.depth_mode = sl.DEPTH_MODE.PERFORMANCE

# Open the camera
err = zed.open(init_params)
if err != sl.ERROR_CODE.SUCCESS:
    print("Camera failed to open:", err)
else:
    print("ZED 2i Camera successfully initialized!")

# Capture a single frame
image = sl.Mat()
if zed.grab() == sl.ERROR_CODE.SUCCESS:
    zed.retrieve_image(image, sl.VIEW.LEFT)
    print("Frame captured successfully!")

zed.close()
```

---

8. Additional Applications of ZED 2i Camera

The ZED 2i is designed for a wide range of **AI, robotics, and vision applications**, including:

- Autonomous navigation and obstacle detection
- 3D scene reconstruction and SLAM
- Human pose estimation and crowd analytics
- Drone-based mapping and surveillance
- Object detection and spatial AI
- Mixed reality and volumetric video generation

With its robust SDK and AI compatibility (PyTorch, TensorFlow, ROS), it supports both research and industrial use cases.

---

9. Conclusion

This SOP provides a complete walkthrough from **installation to advanced operations** of the ZED 2i stereo camera. Following these steps ensures successful setup, optimal calibration, and efficient use of all ZED utilities. By mastering these procedures, users can leverage the full potential of ZED's stereo vision technology for computer vision, robotics, and AI-driven applications.

---

10. Acknowledgment

I would like to extend my heartfelt gratitude to my mentors and guides at **CDAC (Centre for Development of Advanced Computing)** — **Kaushal Sharma Sir, Harshada Deshpande Ma'am, Aaditya Khapre Sir, Mihir Harne Sir and Aman Shrivastava Sir** — for their invaluable support, technical insights, and encouragement throughout the preparation and testing of this SOP. Their continuous guidance and believing in my capabilities helped me in making this SOP more perfect and according to the required Standard.

---

11. References

1. **Stereolabs ZED SDK Documentation**, *Stereolabs Developers Portal*, https://www.stereolabs.com/docs/

2. **CUDA Toolkit Documentation**, *NVIDIA Developer Portal*, https://developer.nvidia.com/cuda-toolkit

3. **Python 3.10 Official Documentation**, *Python Software Foundation*, https://docs.python.org/3.10/

4. **OpenCV Library Reference**, *OpenCV.org*, https://docs.opencv.org/

5. **PyTorch Framework Guide**, *PyTorch Official Site*, https://pytorch.org/docs/stable/

6. **Microsoft Visual Studio Developer Documentation**, *Microsoft Learn*, https://learn.microsoft.com/en-us/visualstudio

---