LIPS
Sense The World

LIPS
Sense The World

LIPSense™ 3D Body Pose SDK User's Manual

V7.2.1

Although every effort has been made to ensure the accuracy of this manual, errors and inconsistencies may remain. The manufacturer assumes no liability resulting from errors or omission in this manual, even if damage arises from the use of the information. All contents are subject to constant revision to improve its reliability and may be changed without prior notice.

April 2026

Contents

Notes for Programmers 4

I. System Requirements 4

II. Hardware Compatibility 5

III. External Dependency Issues 5

IV. Python Support 6

V. Unity Support (Windows Only) 6

LIPSense™ 3D Body Pose SDK User's Manual 7

1. Overview 8

1.1 Camera Accessories 11

2. Detection Requirements 13

2.1 Hardware Requirements 13

2.2 Pose Requirements 15

3. SDK Structure 17

Windows 2

4. SDK Installation 18

5. Example Application 21

6. License File Settings 26

7. C++ Source Code Compilation 28

Linux (NVIDIA Jetson Platform) 33

8. SDK Installation 33

9. Example Application 35

10. License File Settings 37

11. Source Code Compilation 38

Additional Configurations 33

13. Additional Hardware Packages 47

13.1 NVIDIA GPU 47

Supported API Wrappers 33

14. C# 49

14.1 C# Example Compilation Tutorial 49
15. Unity 52
15.1 Unity Configuration Tutorial 52

Notes for Programmers

I. System Requirements

For optimal performance, it is recommended to use the following operating system configuration.

x86 based systems		
Standard version requirements	CPU	6th generation Intel Core™ processor
	RAM	8 GB RAM or above
	GPU	NVIDIA RTX 4060
	Acceleration frameworks	Intel® OpenVINO™, NVIDIA® TensorRT™, Cuda 12.x, cuDNN 9.9.0
Professional version requirements	CPU	8th generation Intel Core™ processor
	RAM	8 GB RAM or above
	GPU	NVIDIA RTX 4060 Ti
	Acceleration frameworks	Intel® OpenVINO™, NVIDIA® TensorRT™, Cuda 12.x, cuDNN 9.9.0
Arm based systems		
NVIDIA® Jetson Xavier™ / AGX Orin™		

II. Hardware Compatibility

The LIPSense™ 3D Body Pose SDK is compatible with the following 3D cameras:

Series	Models
LIPSedge™	AE400 / AE430 / AE450 / AE470
	T225 / T235
	DL
	S315 (Windows Only)

III. External Dependency Issues

The LIPSense™ 3D Body Pose SDK utilizes third-party libraries, including OpenCV. Consequently, the functionality of the LIPSense™ 3D Body Pose SDK depends on the conditions and performance of these external libraries. If any errors occurred during the development process, it is recommended to refer to the release notes of the respective external libraries for debugging purposes.

Library	Release Notes
OpenCV 4.7.0	https://github.com/opencv/opencv/releases/tag/4.7.0
NVIDIA® TensorRT V10.9.0	https://github.com/NVIDIA/TensorRT/releases/tag/v10.9.0
OpenVino V2025.1	https://github.com/openvinotoolkit/openvino/releases/tag/2025.1.0

IV. Python Support

For Linux, use the default Python 3 package in the Ubuntu repository. The Python support for the LIPSense™ 3D Body Pose SDK differs based on the operating system. To use the SDK on Windows, download and install Python 3.12. On Linux, it is recommended to utilize the default Python 3 package available in the Ubuntu repository.

Operating Systems		Supported Python Versions
Windows		3.12
Linux	Ubuntu 22.04	3.10
	Ubuntu 24.04	3.12

V. Unity Support (Windows Only)

Unity support for the LIPSense™ 3D Body Pose SDK is exclusive to **Windows**. We recommend utilizing Unity **2021.3** for optimal compatibility. Proceed to download and install Unity accordingly.

LIPSense™ 3D Body Pose SDK

User's Manual

Welcome to the **LIPSense™ 3D Body Pose SDK User's Manual**! This comprehensive document offers a detailed, step-by-step guide on how to effectively utilize the LIPSense™ 3D Body Pose SDK and set up the development platform on the host PC or laptop.

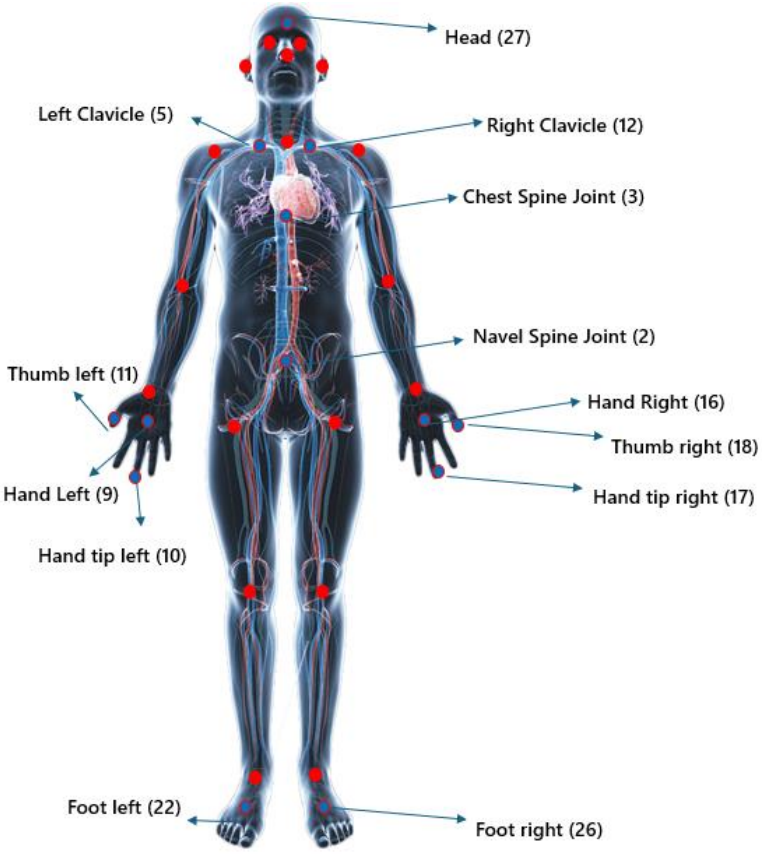
1. Overview

The **LIPSense™ 3D Body Pose SDK v7.2.1** is a high-performance skeletal tracking solution designed for real-time human motion analysis, fully compatible with LIPS Corporation's **LIPSedge ™ AE / DL / T series, and S315**[*Note 1] 3D depth cameras. To accommodate various application needs and system capabilities, the SDK offers 2 **detection versions:** the **Standard version** and **Professional version**.

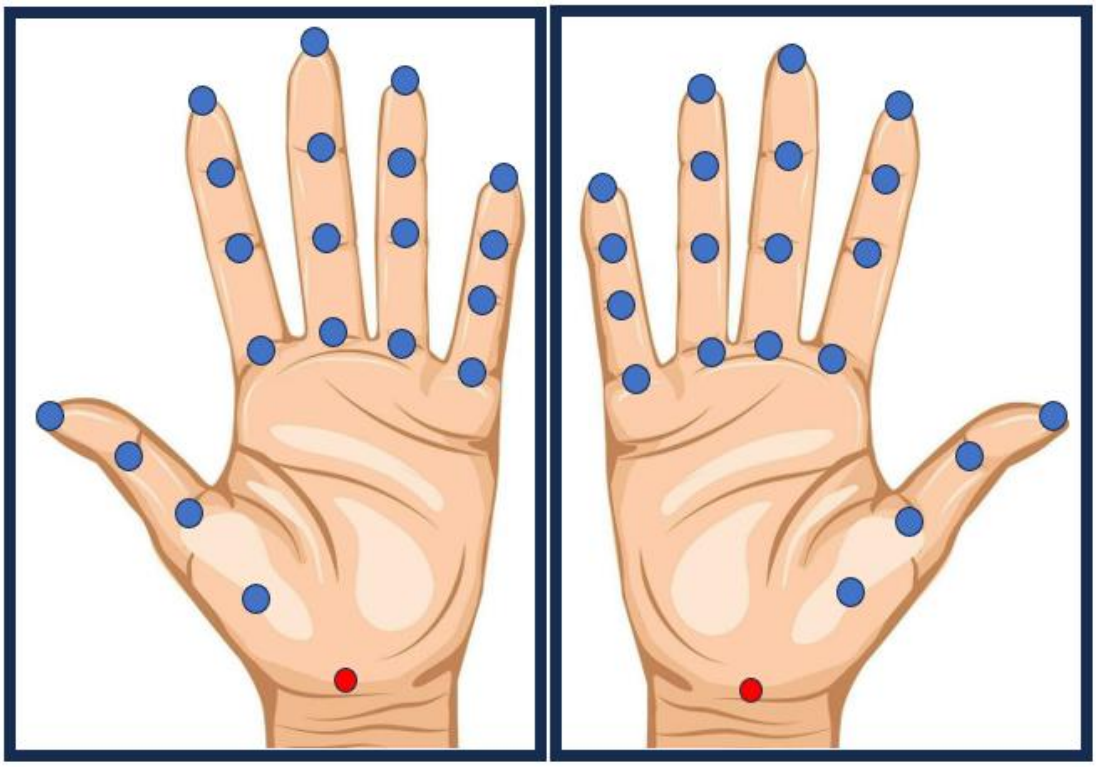
- The **Standard version** offers comprehensive full-body tracking across 32 skeletal key points without the license-based time constraint. This mode is ideal for motion capture, worker safety analytics, and immersive virtual interactions.
- The **Professional version** captures additional 40 key points across both hands—covering 4 joints per finger (including thumbs) and palm centers. It is purpose-built for gesture recognition, sign language interpretation, and precise hand-based control in VR/AR environments. In addition, **selective hand point tracking** is available within Professional version, enabling applications that require combined full-body and hand gesture analysis without overwhelming system resources. Together, these detection modes deliver scalable, high-accuracy skeletal tracking tailored for industrial, embedded, and interactive use cases.

Note: The **LIPSedge ™ S315** is supported by **Windows SDK ONLY**.

[Additional Bodypose key points (32 points)]



[High-precision key points]



Integrated with advanced algorithms, the LIPSense™ 3D Body Pose SDK v7.2.1 delivers superior accuracy, cross-platform flexibility, and enhanced hardware acceleration. This integration utilizes CNN-based deep learning, unified API hardware acceleration, and optimized OpenCV code, ensuring high-performance visualization.

Features

- Full-body skeletal tracking
- Max. joint-tracking capability of 72 joints
  - **Standard version**: 32 key points in body total, including eyes * 2, ears * 2, nose, head, neck, clavicle joints * 2, shoulders * 2, elbows * 2, wrists * 2, hands * 2, tips of the hands * 2, thumbs * 2, pelvis, chest, navel, hips * 2, knees * 2, ankles * 2, feet * 2
  - **Professional version:** additional 40 Finger Joints including 4 joints (knuckle, Middle Joint, Upper Joint) per finger (thumbs included with the base joint added after the palm center), and palm centers for both hands.
- Precise finger tracking functionality
- People-tracking capability for up to 43 people *Note 1
- Proprietary framework for optimal pose detection performance
- Capture of swinging and staggering motions
- Support for Unity and multiple programming languages / platforms
- Support for multiple 3D camera mechanical positions (90° clockwise / 90° counterclockwise / 180°)
- Intel® OpenVINO™ compatibility for Intel® HD graphics inference engine acceleration
- NVIDIA® TensorRT compatibility for NVIDIA® graphics inference engine acceleration
- Compatibility with LIPSedge™ AE series
- Compatibility with LIPSedge™ S series (Windows Only)
- Compatibility with LIPSedge™ DL series
- Compatibility with LIPSedge™ T series
- Compatibility with NVIDIA® Jetson Xavier™ & Orin™

Note:

1. The people tracking capability for LIPSense™ 3D Body Pose SDK depends on the hardware capacity of the developer's PC / laptop. This feature describes the regular capability of LIPSense™ 3D Body Pose SDK.
2. OpenVINO™ is not compatible with NVIDIA Jetson platforms.

1.1 Camera Accessories

Before proceeding with the installation process, ensure the following accessories (sold separately) are prepared:

- LIPSedge™ DL 3D Camera / LIPSedge™ S315 3D Camera / LIPSedge™ AE Series Ruggedized 3D Camera
- Camera tripod
- PC/Laptop with NVIDIA graphics card

For details on LIPSedge™ AE400 / 450, refer to *LIPSedge™ AE400 / 450 Ruggedized 3D Stereo Camera User's Manual*.

For details on LIPSedge™ AE430 / AE470, refer to *LIPSedge™ AE430 / AE470 Ruggdized 3D Stereo Camera User's Manual*.

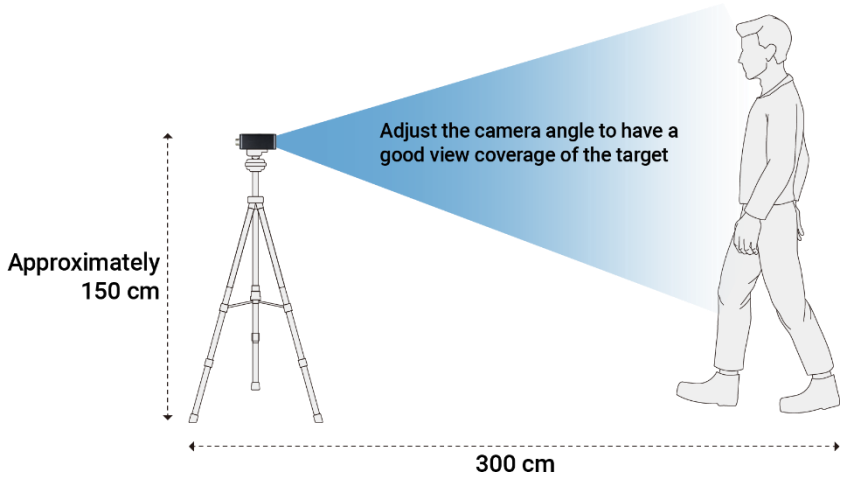
For details on LIPSedge™ DL, refer to LIPSedge™ DL / M3 All-purpose / Embedded 3D ToF Camera User's Manal.

For details on LIPSedge™ T series and S315, the materials will be released in near future.

2. Detection Requirements

2.1 Hardware Requirements

To ensure optimal recognition results, it is important to meet the following requirements:



- **Camera Working Distance:** Maintain an approximate distance of 3 meters (300 cm) between the camera and the recognition target.
- **Camera Position:** Install the camera on a **tripod**, ensuring that its **field of view (FOV) encompasses the full body of the target**. The LIPSense™ 3D Body Pose SDK supports various mechanical positions, including 90° clockwise, counterclockwise, and 180° rotations.
- **Camera Height:** Position the camera at **around 150 cm** above the ground.
- **Illumination:** Ensure sufficient lighting during skeleton detection. AVOID installing the camera in locations with **direct sunlight** or complete darkness.
- **Background:** Keep the background of the recognition target as simple as possible, such as a clean white background, to minimize depth noises.

By complying with these requirements, users can enhance the recognition results and overall system performance. Once all the criteria are fulfilled, connect the camera to the host PC / laptop.

For details on LIPSedge™ AE400 / 450, refer to *LIPSedge™ AE400 / 450 Ruggedized 3D Stereo Camera User's Manual*.

For details on LIPSedge™ AE430 / AE470, refer to *LIPSedge™ AE430 / AE470 Ruggdized 3D Stereo Camera User's Manual*.

For details on LIPSedge™ DL, refer to LIPSedge™ DL / M3 All-purpose / Embedded 3D ToF Camera User's Manal.

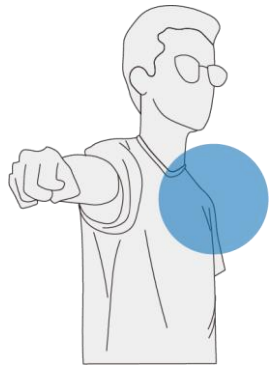
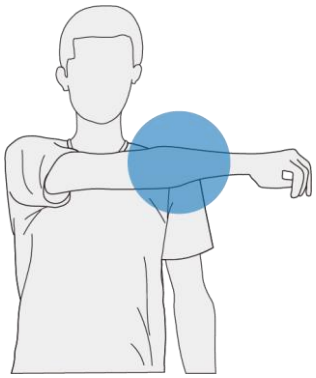
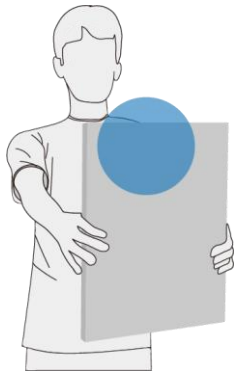
For details on LIPSedge™ T series and S315, the materials will be released in near future.

Following these resources will be helpful in successfully setting up the camera for optimal usage with the LIPSense™ 3D Body Pose SDK.

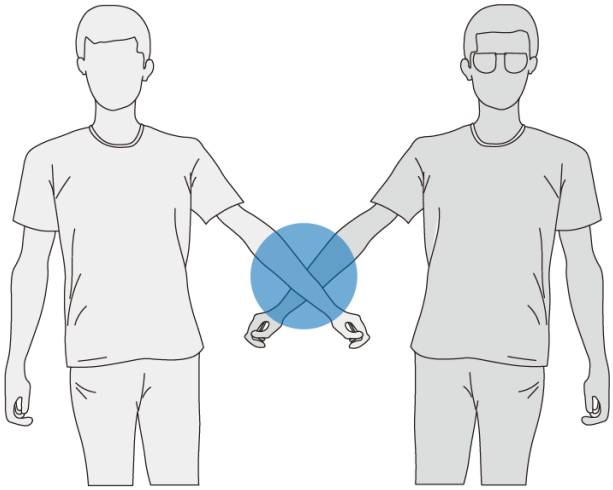
2.2 Pose Requirements

To optimize recognition results, it is recommended to avoid poses where obstacles obstruct or overlap with the recognition target's joints, as this can lead to suboptimal recognition accuracy.

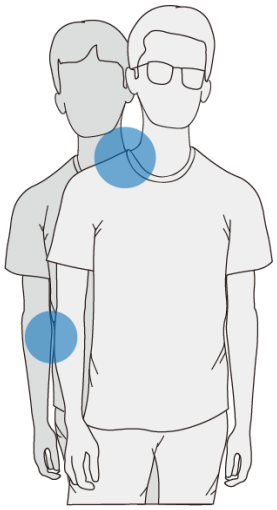
Arm positions (single target)

In these scenarios, when the shoulder joints are obstructed by another limb or an obstacle, it can result in inaccurate tracking outcomes.		
		

Arm positions (multiple target)

In this case, one person's arm is blocking another person's wrist.


In these cases, the joints are overlapping with each other.

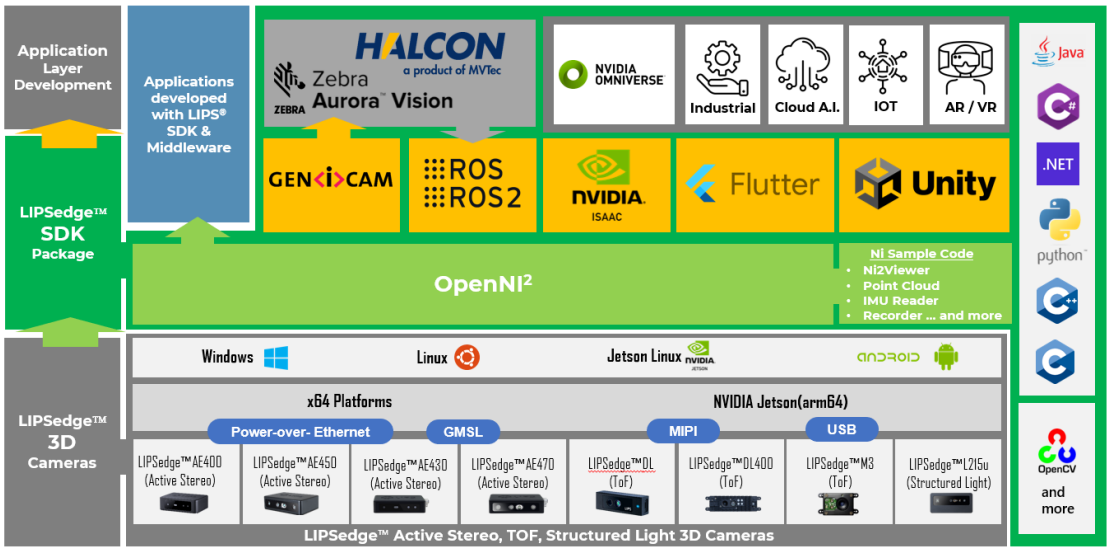


3. SDK Structure

LIPS 3D camera / SDK offers a system for developing depth-sensing applications. As the LIPS system architecture illustrates, the system is comprised of the hardware layer and the software layer.

The hardware layer oversees data capture, transfer, and processes.

In the software layer, the captured data is fetched by the LIPS SDK (Software Development Kit) on the OS environment. Depending on the project complexity, wrappers and third-party utilities may be engaged before the data is eventually presented in the application layer for business applications.



The core of the system, the LIPS SDK, is comparable to a toolbox full of software modules comprised of middleware, libraries, wrappers and API, and miscellaneous programming languages / platforms for application development. With extensive wrapper support, LIPS SDK enables developers to access bottom layer data with APIs, thus eliminating the hassle of changing third-party functions. The result is a highly effective project scoping, monitoring, and execution workflow compatible with the fast-pacing AIoT market and machine vision demands.

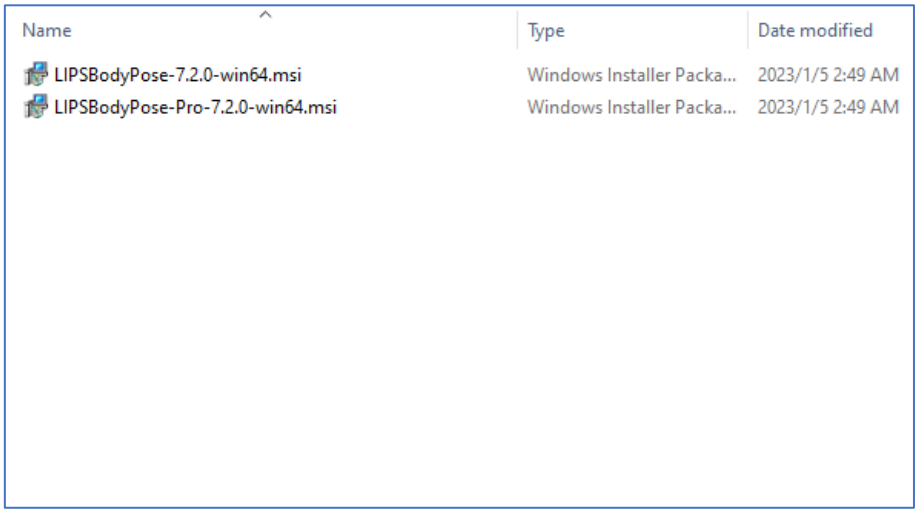
LIPS
Sense The World

Windows

4. SDK Installation

LIPS Corp. has released versions of the LIPSense™ 3D Body Pose SDK for specific systems on the official LIPS Corp. websites. Download and install the LIPSense™ 3D Body Pose SDK according to the host platform.

1. Unzip the downloaded file and double-click the **installation file** to install the SDK.
   - For the **Standard** version, click **LIPSBodyPose-[Version No.]-[Operating system].msi.**
   - For the **Professional** version, click **LIPSBodyPose-Pro-[Version No.]-[Operating system].msi**

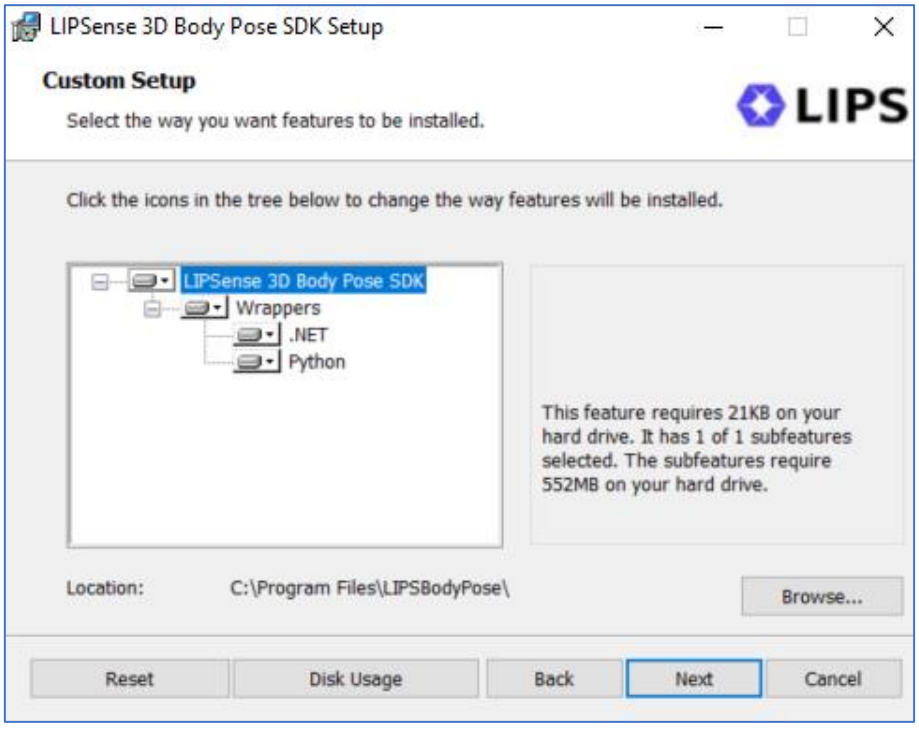


2. Click **Next**.

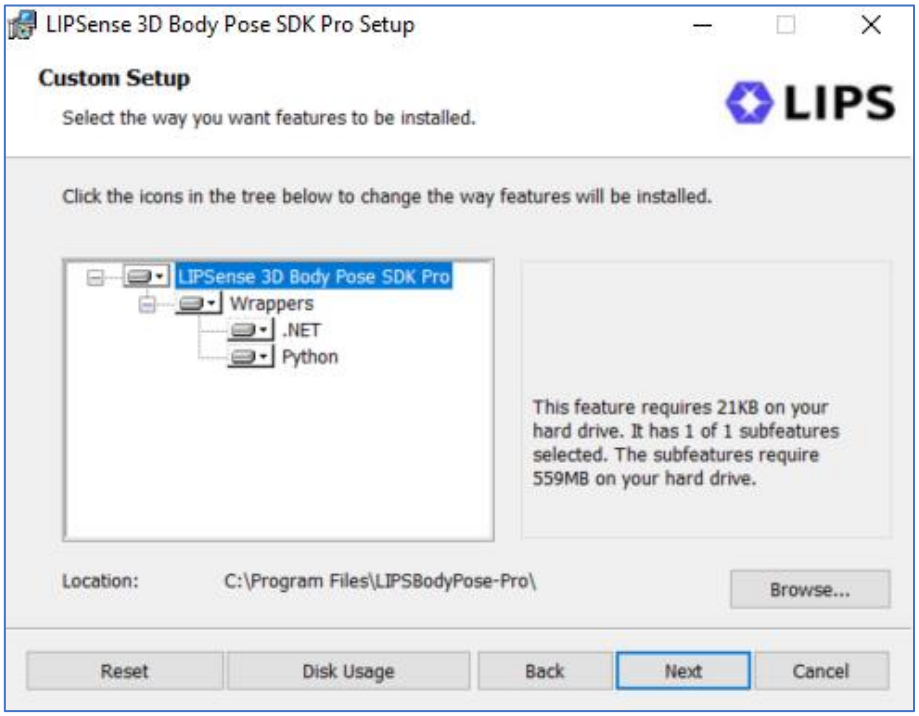
3. Review the license agreement and click **Next**.

4. Choose an **installation directory** for the LIPSense™ 3D Body Pose package, including:
   - Neural engine plugins
   - Wrappers

[Standard version]



[Professional version]

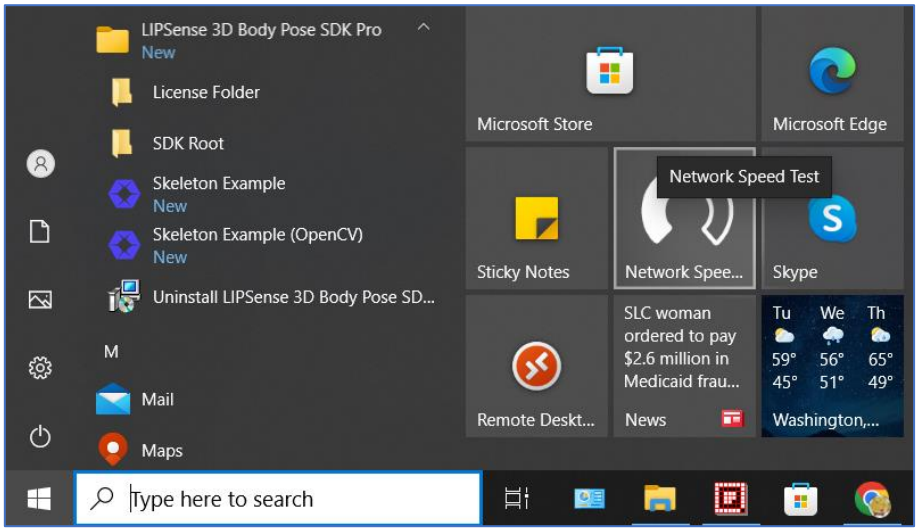


5. Click **Next**.

6. Click **Install**.

7. Click **Finish**.

8. Once the installation is completed, developers can access LIPSense™ 3D Body Pose SDK though the **Windows Start Menu**.



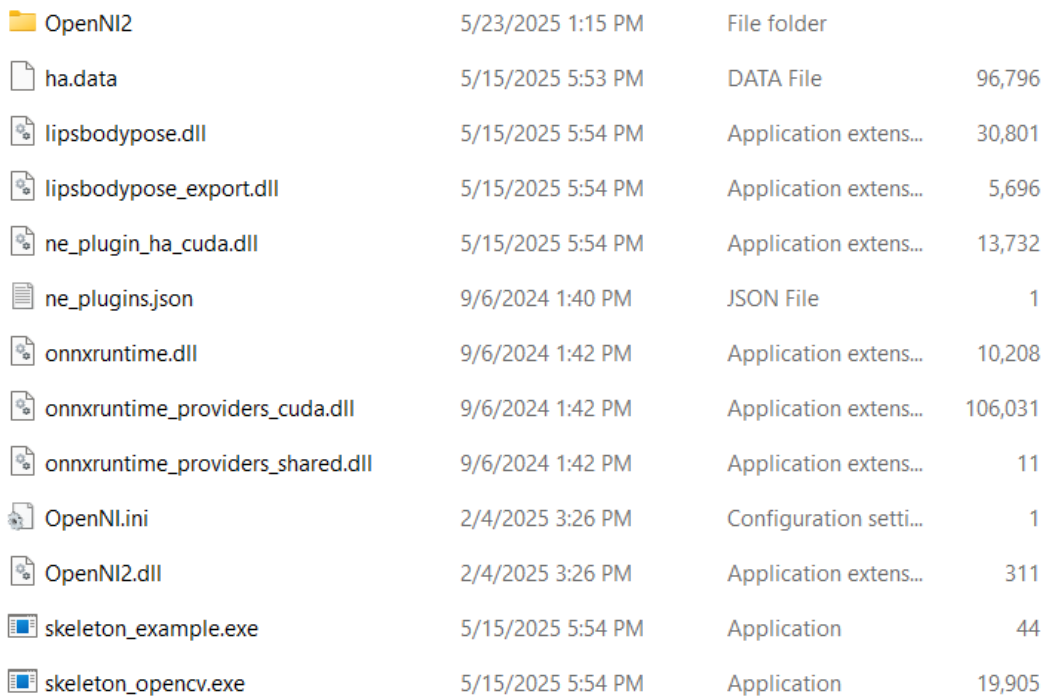
After installing LIPSense™ 3D Body Pose SDK, additional configuration is required to ensure smooth operation of the SDK. For more information, refer to the **Additional Configurations** section.

5. Example Application

LIPS Corp. provides a **Skeleton Example** application to demonstrate the interaction possibilities of LIPSense™ 3D Body Pose SDK. Before launching the example application, make sure that the camera is properly connected and it is recommended to use the following operating system and compiler:

- Windows 11 64-bit system or later
- Microsoft Visual Studio 2022 with Microsoft Visual C++ Redistributable 2022

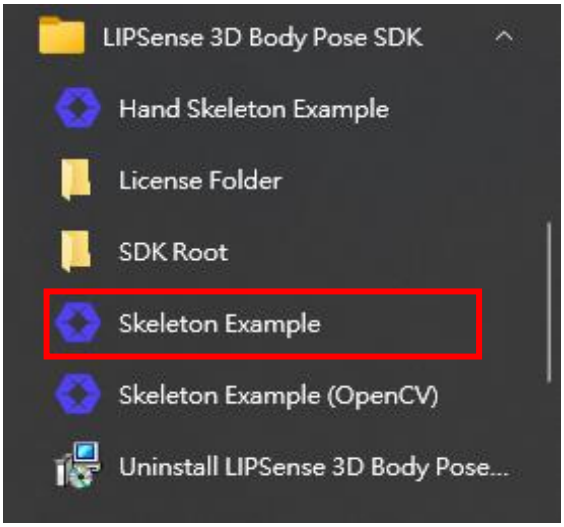
Note: For **LIPSedge™ AE400 / AE450** users, make sure the properly configured **network.json** (camera network configuration file) is in the same folder as skeleton_example.exe to start the application. The default location of LIPSense™ 3D Body Pose SDK is C:\Program Files\LIPSBodyPose\bin.



OpenNI2	5/23/2025 1:15 PM	File folder	
ha.data	5/15/2025 5:53 PM	DATA File	96,796
lipsbodypose.dll	5/15/2025 5:54 PM	Application extens...	30,801
lipsbodypose_export.dll	5/15/2025 5:54 PM	Application extens...	5,696
ne_plugin_ha_cuda.dll	5/15/2025 5:54 PM	Application extens...	13,732
ne_plugins.json	9/6/2024 1:40 PM	JSON File	1
onnxruntime.dll	9/6/2024 1:42 PM	Application extens...	10,208
onnxruntime_providers_cuda.dll	9/6/2024 1:42 PM	Application extens...	106,031
onnxruntime_providers_shared.dll	9/6/2024 1:42 PM	Application extens...	11
OpenNI.ini	2/4/2025 3:26 PM	Configuration setti...	1
OpenNI2.dll	2/4/2025 3:26 PM	Application extens...	311
skeleton_example.exe	5/15/2025 5:54 PM	Application	44
skeleton_opencv.exe	5/15/2025 5:54 PM	Application	19,905

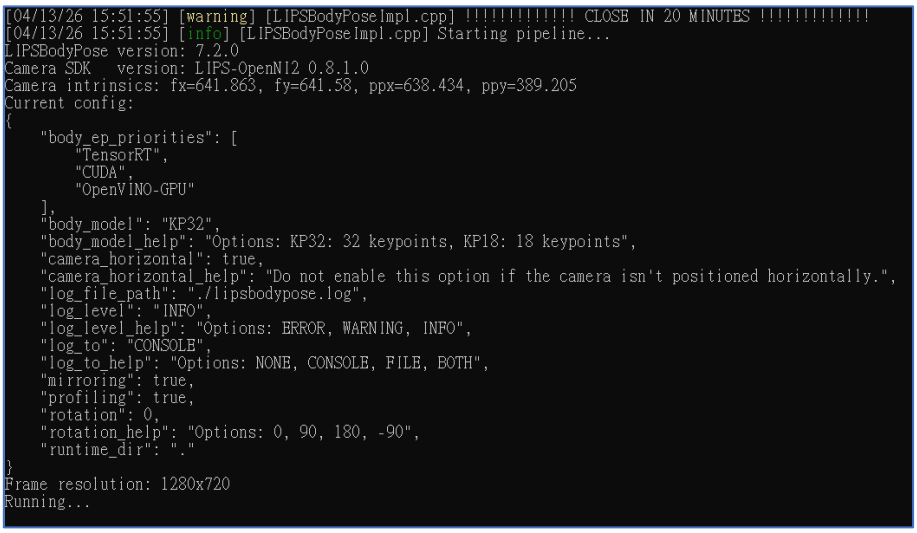
[OpenCV Program]

1. From Windows Start Menu, find LIPSense™ 3D Body Pose SDK and start Skeleton Example.

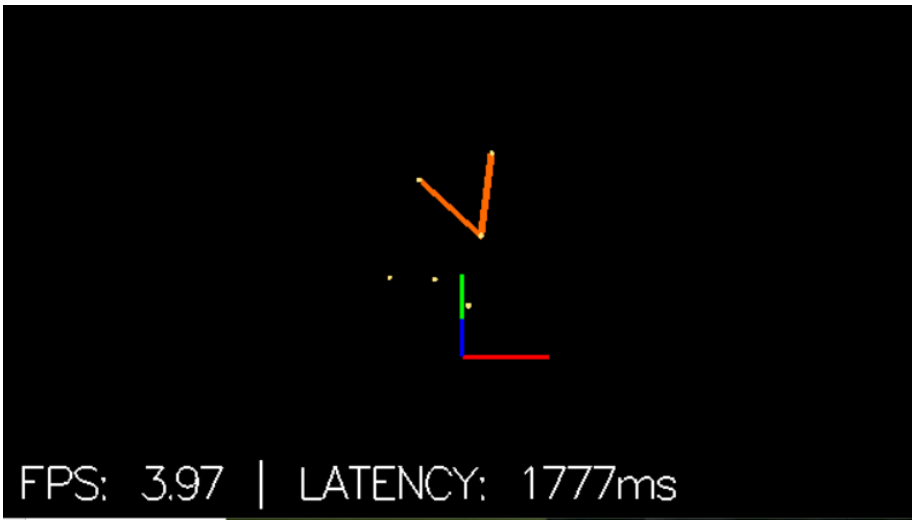


2. 2 windows pop up:
   - **Console Window**: Displays the camera information, error logs and status.
   - **3D Window** : Displays the live skeleton detection results in 3D images.

[Console Window]



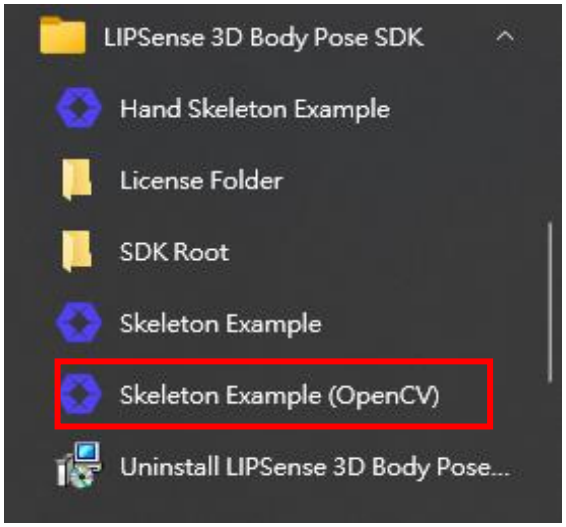
[3D Windows]



[OpenCV Program]

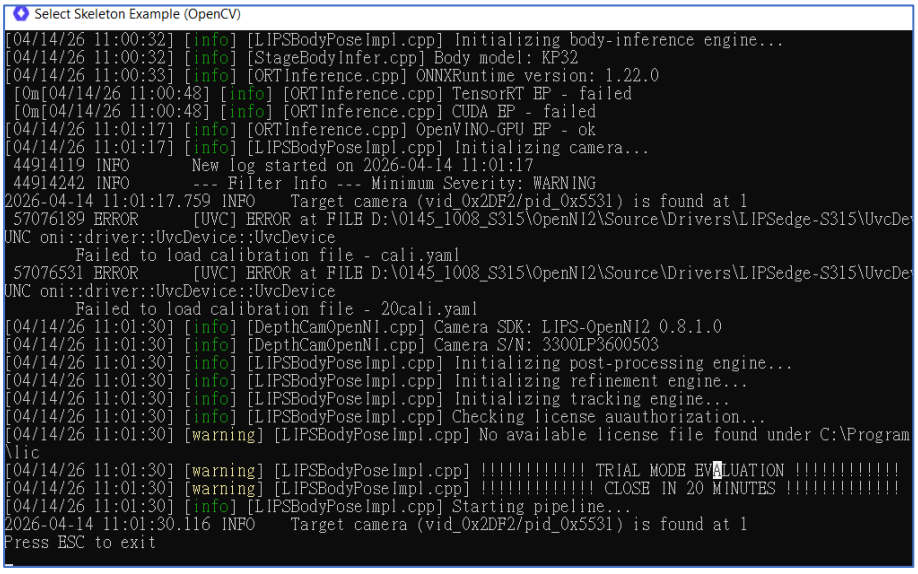
The Skeleton Example (OpenCV) program demonstrates the results that LIPSense 3D Body Pose SDK working with OpenCV

1. From Windows Start Menu, find LIPSense™ 3D Body Pose SDK and start **Skeleton Example(OpenCV).**

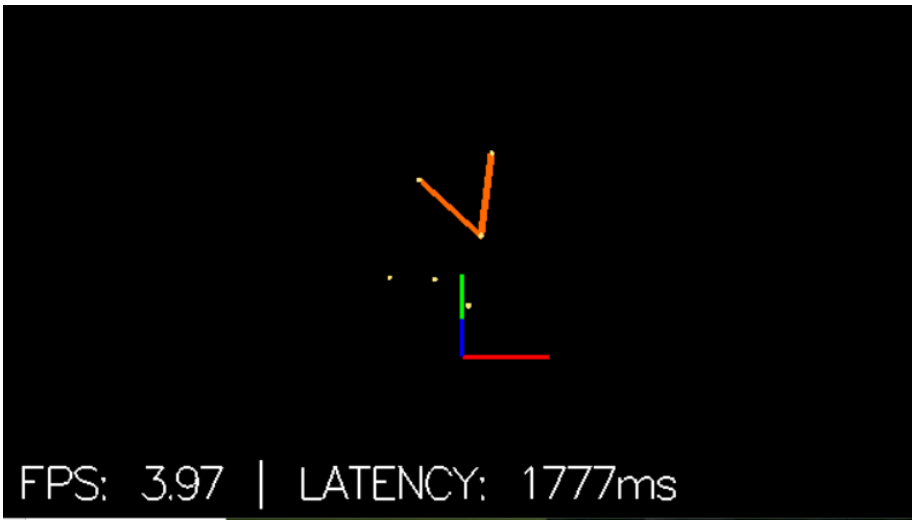


3. 2 windows pop up:
   - **Console Window**: Displays the camera information, error logs and status.
   - **3D Windows**: Displays the live skeleton detection results in 3D images.

[Console Window]



[3D Window]



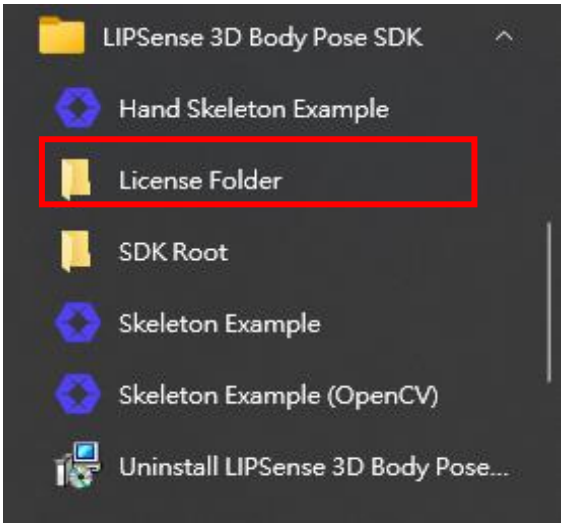
Note: Certain poses cause less optimal recognition results. Avoid those poses during the motion tracking process. For details, refer to *2.2 Pose Requirement, LIPSense™ 3D Body Pose SDK User's Guide.*

6. License File Settings

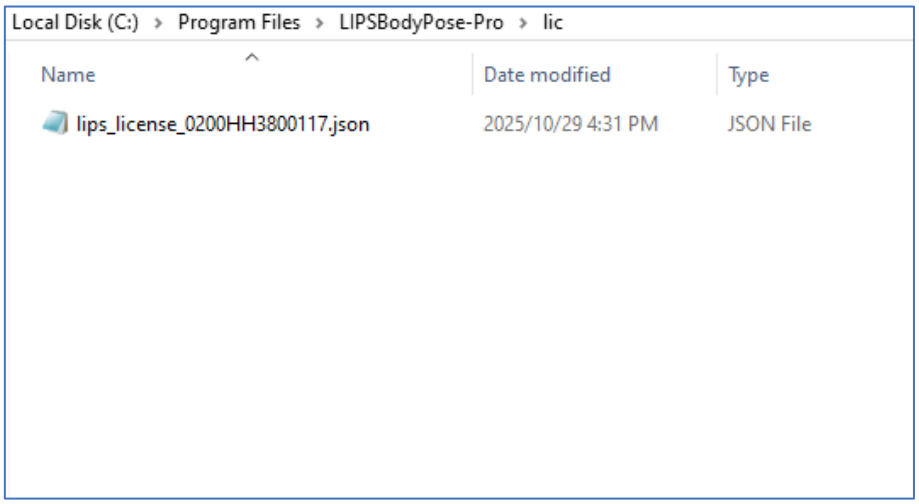
The example application of LIPSense™ 3D Body Pose SDK has a 20-minute time limit in Trial mode. To remove the time limit, contact info@lips-hci.com to obtain a license file. Next, follow the instructions to add the license file's location to the local environment.

1. Obtain the license file from LIPS Corp.

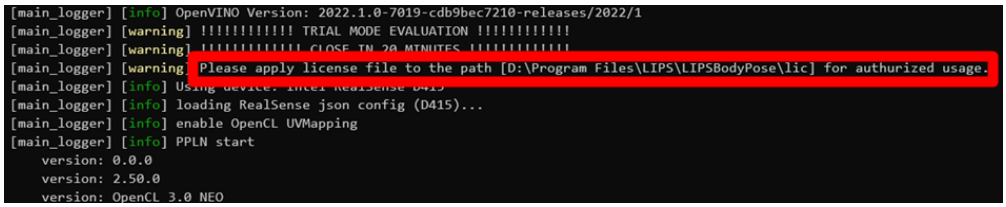
2. From **Windows Start Menu**, select **License Folder**.



3. Move the license file provided by LIPS Corp. to the designated location.
   Installation Directory \LIPSBodyPose\lic



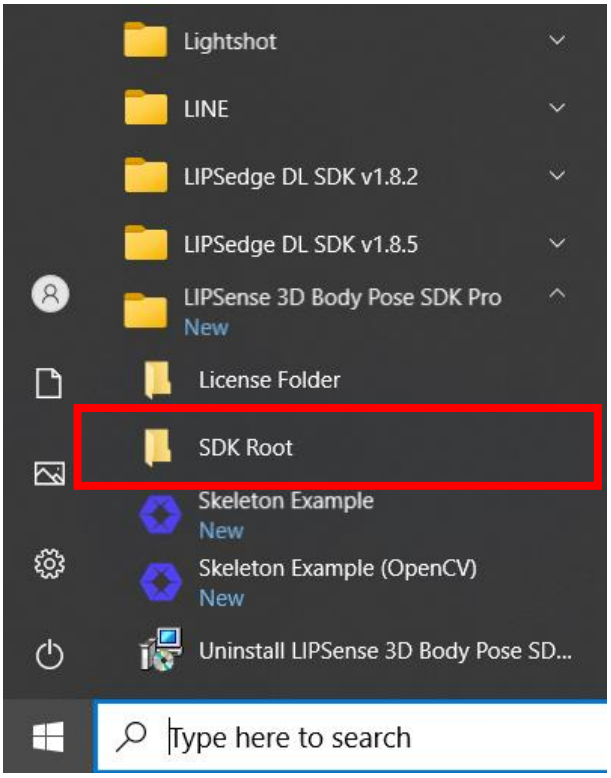
4. If the license file is not found, a warning message will appear in the program's console window indicating the location to store the license file. Obtain the license file and follow steps 1-2 to install it.



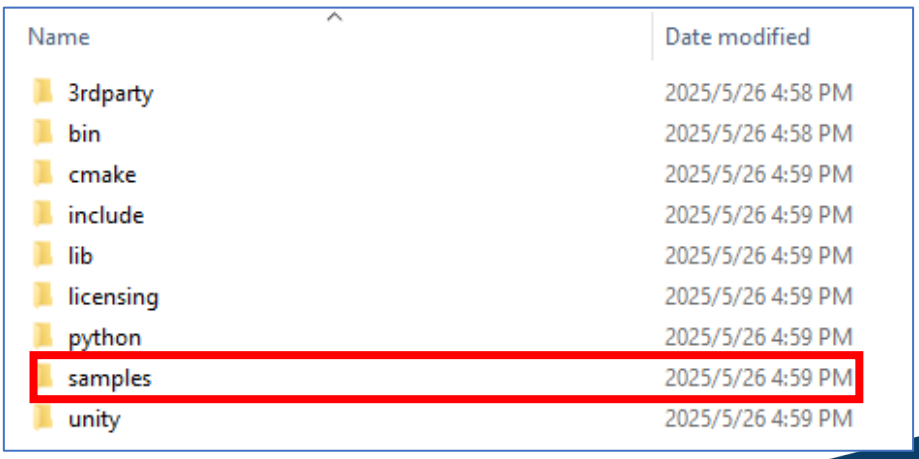
7. C++ Source Code Compilation

LIPS Corp. provides a **source code package** of the example application for developers who wish to customize it. This package allows developers to add, remove, or modify features and compile the source code based on their individual development needs.

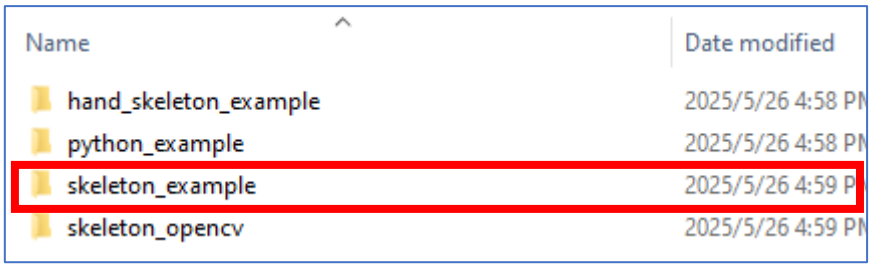
1. From **Windows Start Menu**, navigate to the location of **LIPSense™ 3D Body Pose SDK's location** and select "**SDK Root**".



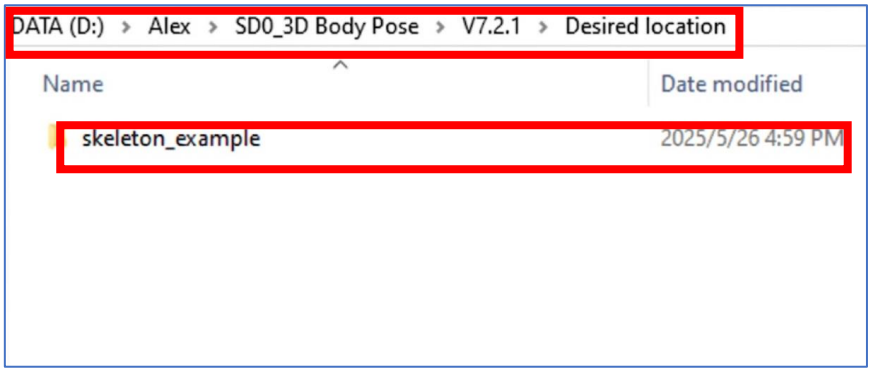
2. Click to enter the samples folder



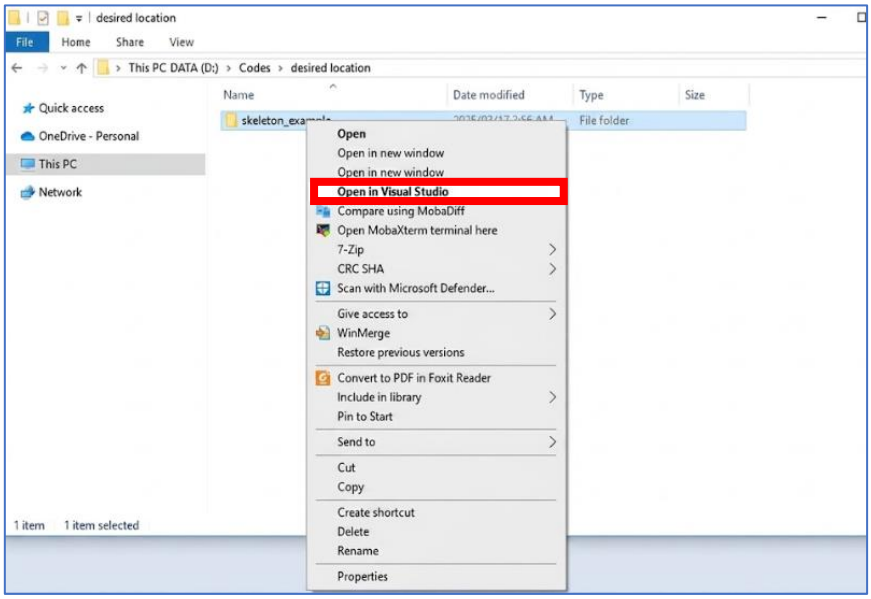
3. In the samples folder, a **skeleton_example** folder is present.



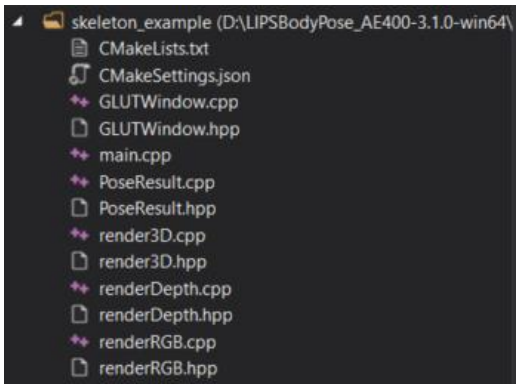
4. From there, copy the "**skeleton_example"** folder to the desired location.



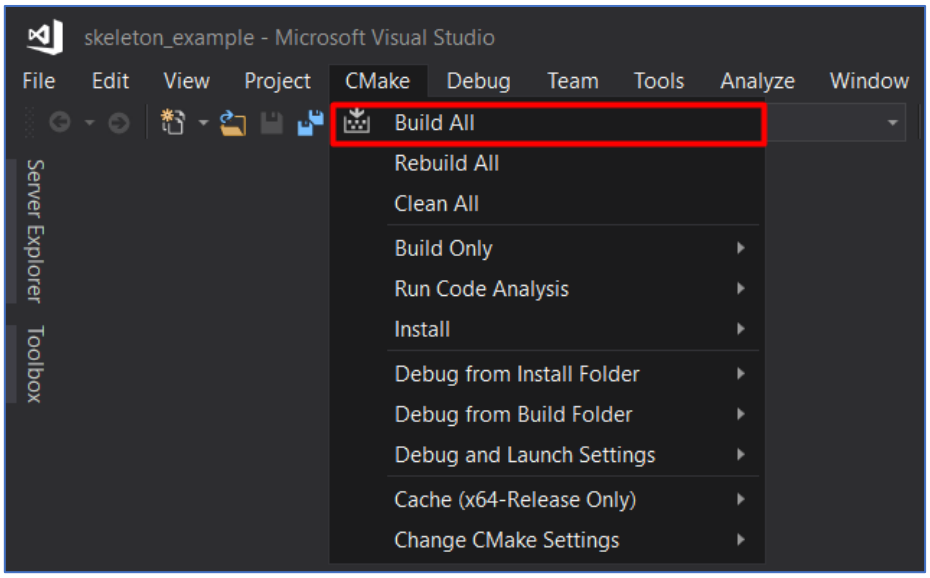
5. Right click **skeleton_example**, and select **Open in Visual Studio**.



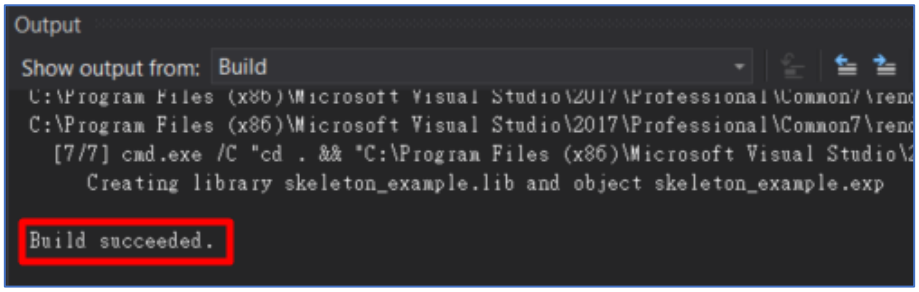
Note: Make sure the **skeleton_example** folder appears in the upper right menu of the Microsoft Visual Studio.



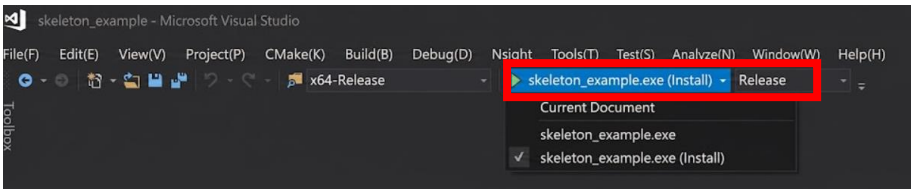
6. From the top bar, select **Cmake > Build All** to compile the **skeleton_example.**



7. Make sure the **Build succeeded** message appears in the Output window to confirm a successful compilation.



8. From the top bar, click the dropdown list for Select Startup Items and choose skeleton_example.exe (Installation).



9. Click the **Arrow sign** to execute the built example application.

LIPS
Sense The World

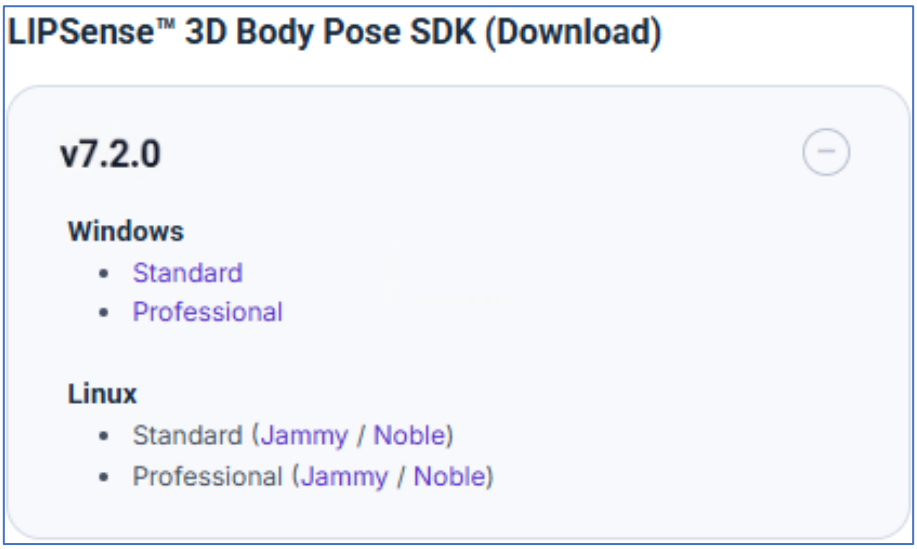
Linux (NVIDIA Jetson Platform)

8. SDK Installation

To access the SDK, visit LIPS®Developer Documentation and procure the package for the preferred version, framework, and compatible operating system.

1. Visit LIPS®Developer Documentation.

2. Select the SDK version compatible with the local environment.



3. Install the downloaded file.

```
sudo apt install ./lipsbodypose_<version>-<environment>.deb
```

```
lips@lips-Inspiron-7570:~$ sudo apt install ./lipsbodypose_7.2.0-jammy_amd64.deb

Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Note, selecting 'lipsbodypose' instead of './lipsbodypose_7.2.0-jammy_amd64.deb'
The following additional packages will be installed:
  intel-opencl-icd libcanberra-gtk-module libcanberra-gtk0 libclang-cpp12
  libglew-dev libglew2.2 libigc1 libigdfcl1 libllvm12 libllvmspirvlib12
  libopencl-clang12
Suggested packages:
  glew-utils
The following NEW packages will be installed:
  intel-opencl-icd libcanberra-gtk-module libcanberra-gtk0 libclang-cpp12
  libglew-dev libglew2.2 libigc1 libigdfcl1 libllvm12 libllvmspirvlib12
  libopencl-clang12 lipsbodypose
0 upgraded, 12 newly installed, 0 to remove and 32 not upgraded.
Need to get 45.1 MB/602 MB of archives.
After this operation, 2777 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

After installing LIPSense™ 3D Body Pose SDK, additional configuration is required for smooth operation of LIPSense™ 3D Body Pose SDK. For details, refer to Additional Configurations.

9. Example Application

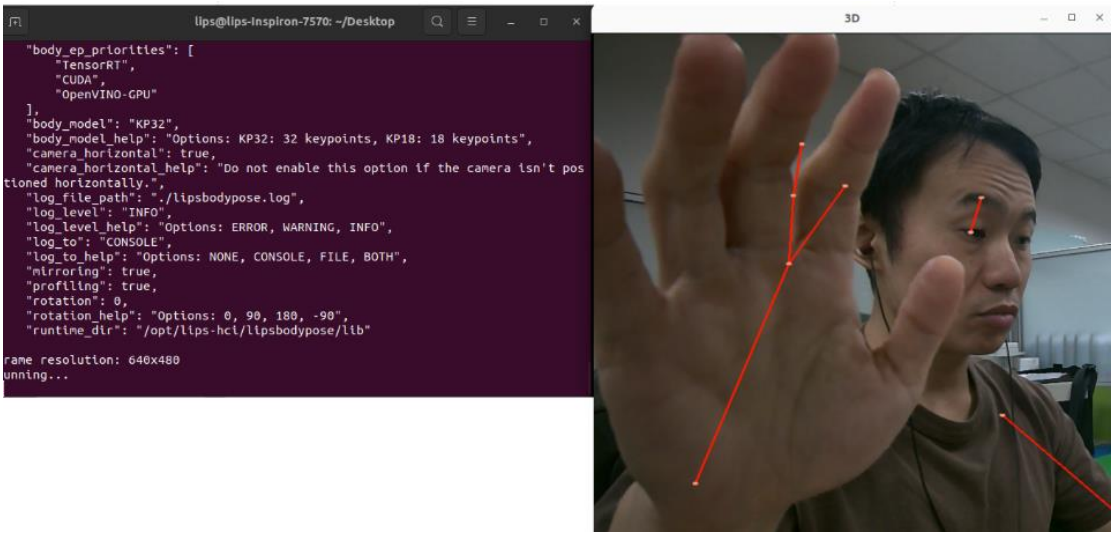
Once installed, **LIPSense™ 3D Body Pose SDK** and its configuration files will be available in local system at **/opt/lips-hci/lipsbodypose**.

LIPS Corp. provides a **Skeleton Example** application that demonstrates the interaction possibilities of LIPSense™ 3D Body Pose SDK. Before launching the example application, ensure that the camera is properly connected. **LIPSense Skeleton Example** is accessible in the **Applications** section located at the bottom left corner.

1. Start **LIPSense™ 3D Body Pose**

```
lips@lips-Inspiron-7570:~/Desktop$ /opt/lips-hci/lipsbodypose/bin/skeleton_examp
le
Initializing...
[04/15/26 14:11:19] [info] [LIPSBodyPoseImpl.cpp] LIPSBodyPose SDK v7.2.0
[04/15/26 14:11:19] [info] [LIPSBodyPoseImpl.cpp] Initializing body-inference en
gine...
[04/15/26 14:11:19] [info] [StageBodyInfer.cpp] Body model: KP32
[04/15/26 14:11:19] [info] [ORTInference.cpp] ONNXRuntime version: 1.22.0
[04/15/26 14:11:19] [info] [ORTInference.cpp] TensorRT EP - failed
[04/15/26 14:11:19] [info] [ORTInference.cpp] CUDA EP - failed
[04/15/26 14:11:25] [info] [ORTInference.cpp] OpenVINO-GPU EP - ok
[04/15/26 14:11:25] [info] [LIPSBodyPoseImpl.cpp] Initializing camera...
2026-04-15 14:11:25.977 INFO    DL Module2 Driver INITIALIZED
2026-04-15 14:11:38.027 INFO    Using OpenNI2 Module Device!
2026-04-15 14:11:38.028 INFO    SDK Build Version: 2.4.5.0_v1.9.2/Release
2026-04-15 14:11:38.028 INFO    SDK Build    Time: Jan 22 2025.18:21:45
2026-04-15 14:11:38.028 INFO    SDK Build    Info: dev-2450-rc@be6cb5ac
2026-04-15 14:11:38.191 INFO    Depth camera VID:PID = 2DF2:0213:0007 at video4
2026-04-15 14:11:38.344 INFO    RGB camera VID:PID = 2DF2:0215:0002 at video2
2026-04-15 14:11:38.344 INFO    Local config not found. Try driver installed pat
```

2. 4 windows pop up: Four windows will pop up:
   - **Console Window**: Displays camera information, error logs, and status.
   - **RGB Window**: Displays live skeleton detection results in RGB images.
   - **3D Window**: Displays live skeleton detection results in 3D images.



Note:

1. To achieve optimal recognition results during the motion tracking process, it is advisable to avoid certain poses. For more information on pose requirements, refer to *Section 2.2 Pose Requirements, LIPSense™ 3D Body Pose SDK User's Guide*.
2. For detailed information about the core functions of the program, refer to *8. Core Functions, Windows, LIPSense™ 3D Body Pose SDK User's Guide*.

---

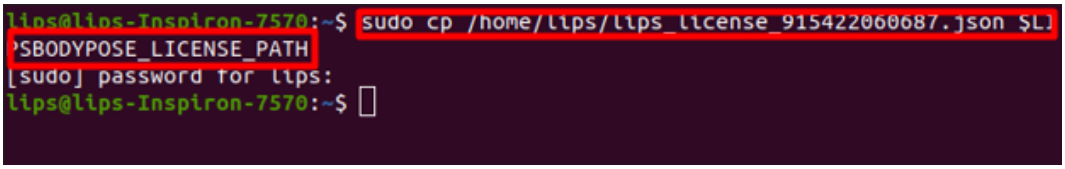
10. License File Settings

LIPSense™ 3D Body Pose SDK's example application comes with a 20-minute time limit in the Trial mode. To remove the time limit, contact info@lips-hci.com to obtain a license file. Next, follow the instructions to add the license file's location to the local environment.

1. Obtain the license file from LIPS Corp.

2. Move the license file provided by LIPS Corp. to the designated location.
   Location: /opt/lips-hci/lipsbodypose/lic

```
sudo cp [Current ocation of the license file]
$LIPSBODYPOSE_LICENSE_PATH
```

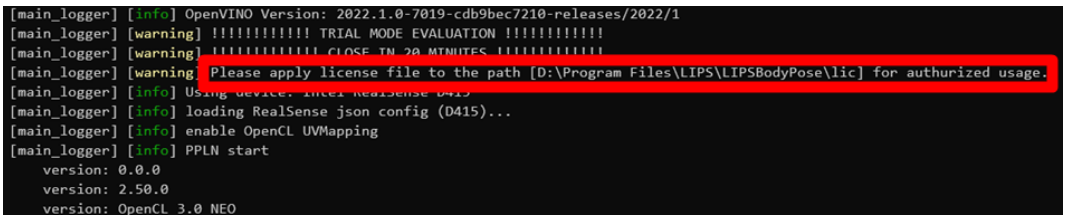


Tip: When specifying the location of the license file, developers have the option to use the abbreviated form $LIPSBODYPOSE_LICENSE_PATH.

---

3. Verify that the license file is present in the specified location.

4. If the license file is absent, a warning message will appear in the program's console window indicating the location to store the license file. Obtain the license file and follow steps 1 and 2 to install the license file.

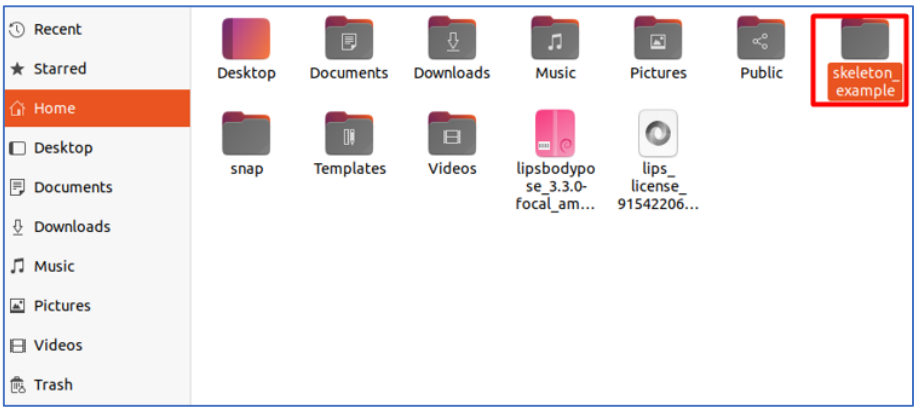


11. Source Code Compilation

For developers who would like to customize the application, LIPS Corp. provides the **source code package** of the example application. This package enables developers to add, remove, or modify features and compile the source code according to their individual development needs.

Note that Cmake tools are required for this process. Make sure **Cmake** is installed on the host PC/laptop.

1. Copy the **skeleton_example** folder from the root location of LIPSense™ 3D Body Pose SDK to the desired location.



Tip: For Terminal users, when specifying the root location of LIPSense™ 3D Body Pose SDK, it is possible to use the abbreviated $LIPSBODYPOSE_SDK_ROOT form.

---

2. Install the **Cmake** tools.

```
sudo apt-get install -y cmake build-essential
```

```
lips@lips-Inspiron-7570:~$ sudo apt-get install -y cmake build-essential
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu cmake-data dpkg-dev
  fakeroot g++ g++-9 gcc gcc-9 libalgorithm-diff-perl
  libalgorithm-diff-xs-perl libalgorithm-merge-perl libasan5 libatomic1
  libbinutils libc-dev-bin libc6-dev libcrypt-dev libctf-nobfd0 libctf0
  libcurl4 libfakeroot libgcc-9-dev libitm1 libjsoncpp1 liblsan0 libquadmath0
  librhash0 libstdc++-9-dev libtsan0 libubsan1 linux-libc-dev make
  manpages-dev
Suggested packages:
  binutils-doc cmake-doc ninja-build debian-keyring g++-multilib
  g++-9-multilib gcc-9-doc gcc-multilib autoconf automake libtool flex bison
  gcc-doc gcc-9-multilib gcc-9-locales glibc-doc libstdc++-9-doc make-doc
The following NEW packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu build-essential cmake
  cmake-data dpkg-dev fakeroot g++ g++-9 gcc gcc-9 libalgorithm-diff-perl
  libalgorithm-diff-xs-perl libalgorithm-merge-perl libasan5 libatomic1
  libbinutils libc-dev-bin libc6-dev libcrypt-dev libctf-nobfd0 libctf0
  libcurl4 libfakeroot libgcc-9-dev libitm1 libjsoncpp1 liblsan0 libquadmath0
  librhash0 libstdc++-9-dev libtsan0 libubsan1 linux-libc-dev make
```

3. In a folder of choice, **create a folder** for the compiled application and go to the folder.

```
cd [Desired folder location]/

mkdir [Compiled application location] && cd [Compiled application location]
```

```
lips@lips-Inspiron-7570:~$ cd skeleton_example/
lips@lips-Inspiron-7570:~/skeleton_example$ mkdir build && cd build
lips@lips-Inspiron-7570:~/skeleton_example/build$
```

4. Run **Cmake**

```
cmake ..
```

```
lips@lips-Inspiron-7570:~/skeleton_example/build$ cmake ..
-- The CXX compiler identification is GNU 9.3.0
-- Check for working CXX compiler: /usr/bin/c++
-- Check for working CXX compiler: /usr/bin/c++ -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Found OpenCV: /opt/lips-hci/lipsbodypose/3rdparty/opencv (found version "4.5.
3")
CMake Warning (dev) at /usr/share/cmake-3.16/Modules/FindOpenGL.cmake:275 (messa
ge):
  Policy CMP0072 is not set: FindOpenGL prefers GLVND by default when
  available.  Run "cmake --help-policy CMP0072" for policy details.  Use the
  cmake_policy command to set the policy and suppress this warning.

  FindOpenGL found both a legacy GL library:

    OPENGL_gl_LIBRARY: /usr/lib/x86_64-linux-gnu/libGL.so

  and GLVND libraries for OpenGL and GLX:

    OPENGL_opengl_LIBRARY: /usr/lib/x86_64-linux-gnu/libOpenGL.so
    OPENGL_glx_LIBRARY: /usr/lib/x86_64-linux-gnu/libGLX.so
```

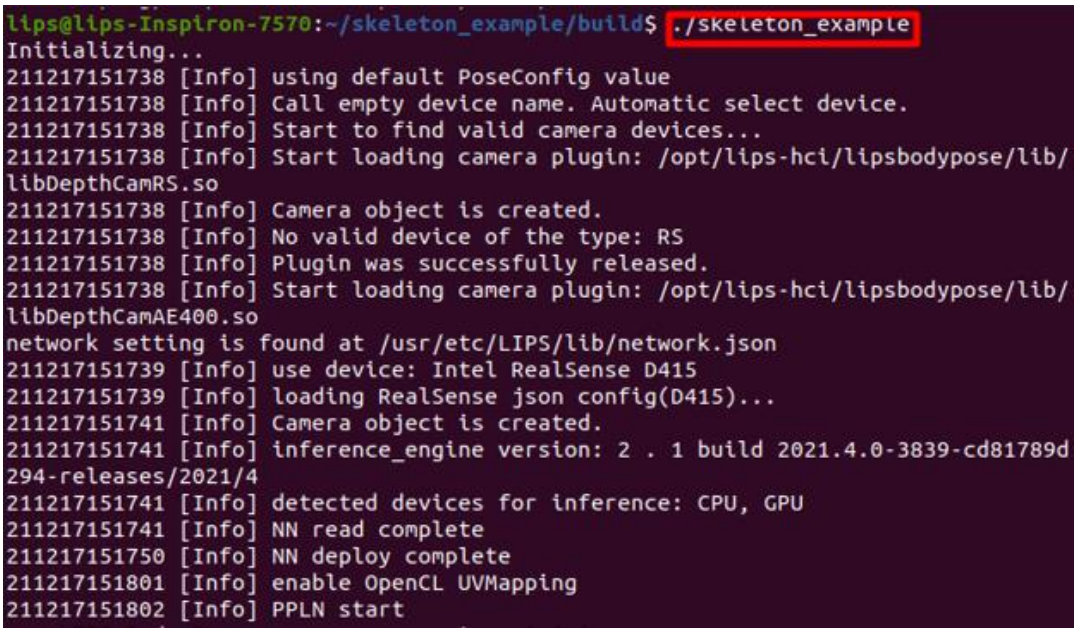
5. Compile the example application.

```
make
```

```
lips@lips-Inspiron-7570:~/skeleton_example/build$ make
Scanning dependencies of target skeleton_example
[ 14%] Building CXX object CMakeFiles/skeleton_example.dir/GLUTWindow.cpp.o
[ 28%] Building CXX object CMakeFiles/skeleton_example.dir/PoseResult.cpp.o
[ 42%] Building CXX object CMakeFiles/skeleton_example.dir/main.cpp.o
[ 57%] Building CXX object CMakeFiles/skeleton_example.dir/render3D.cpp.o
[ 71%] Building CXX object CMakeFiles/skeleton_example.dir/renderDepth.cpp.o
[ 85%] Building CXX object CMakeFiles/skeleton_example.dir/renderRGB.cpp.o
[100%] Linking CXX executable skeleton_example
[100%] Built target skeleton_example
lips@lips-Inspiron-7570:~/skeleton_example/build$
```

6. The example application compiled by using the following command is accessible through the command.

```
./skeleton_example
```



To access the core functions for application development, refer to the following files:

- For details of the core functions and class diagrams, consult the similar content in *8. Core Functions, Windows, LIPSense™ 3D Body Pose SDK User's Guide*

To access the source code of LIPSense™ 3D Body Pose SDK, navigate to the following directory:

- /opt/lips-hci/lipsbodypose/skeleton_example

To access the SDK header files, navigate to the following directory:

- /opt/lips-hci/lipsbodypose/include/LIPSBodyPose

LIPS
Sense The World

Additional Configurations

13. Additional Hardware Packages

Since LIPSense™ 3D Body Pose SDK interacts with various hardware components, including the graphics card, additional packages may be necessary for the local device to function properly with the SDK. Install these packages after completing the installation of LIPSense™ 3D Body Pose SDK

13.1 NVIDIA GPU

For NVIDIA GPU users, install the packages based on the OS by following the official instructions.

Library	Installation Guide
CUDA Toolkit 12.9	https://developer.nvidia.com/cuda-12-9-0-download-archive
cuDNN 9.9.0	https://developer.nvidia.com/cudnn-9-9-0-download-archive

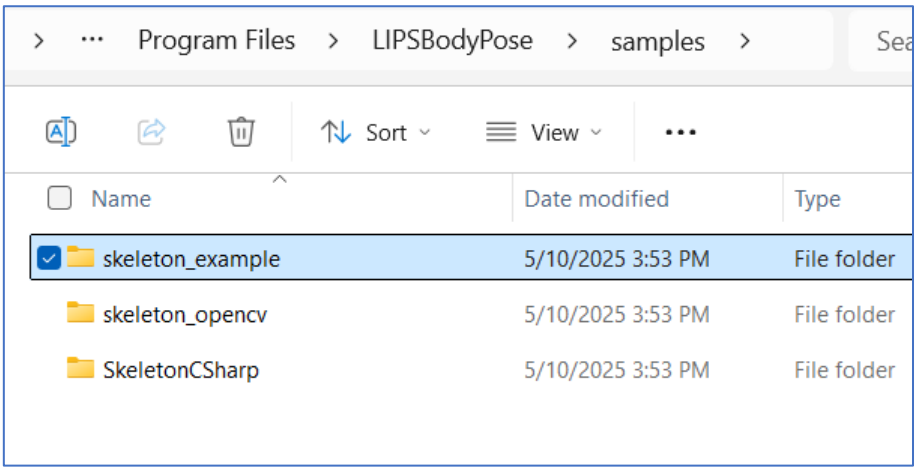
LIPS
Sense The World

Supported API Wrappers

14. C#

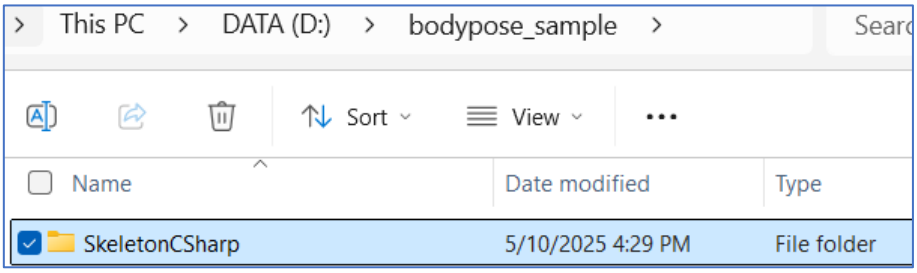
14.1 C# Example Compilation Tutorial

1. After installlation go to C:\Program Files\LIPSBodyPose\samplesCopy SkeletonCSharp folder

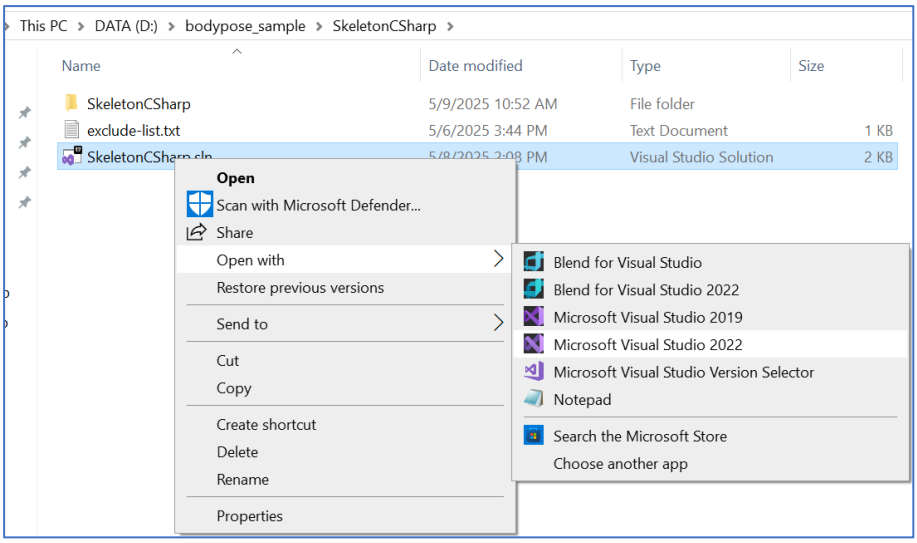


\

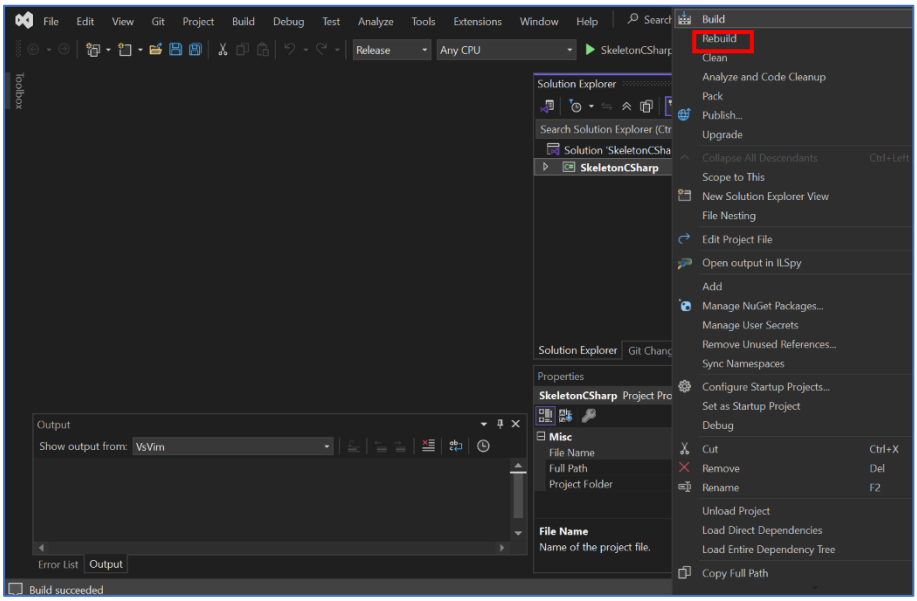
2. Copy the folder into your targeted folder in this example bodypose_sample



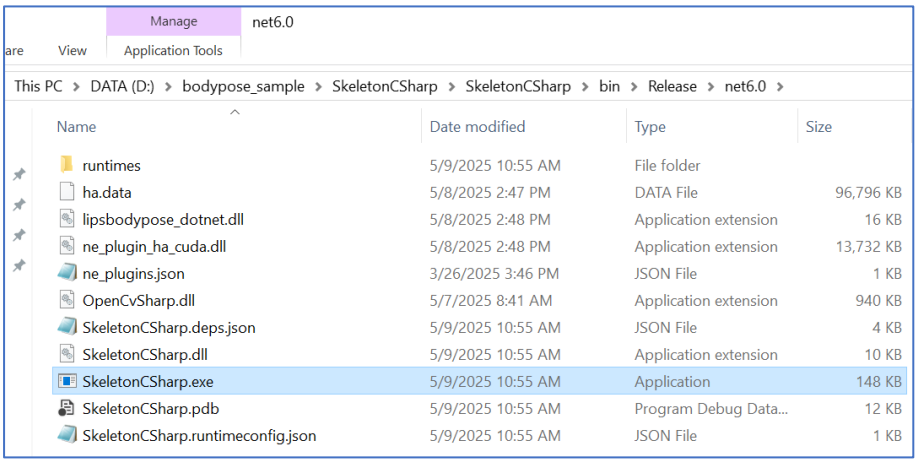
3. Navigate to the copied C# sample folder and launch the sln file with Visual Studio 2022



4. Compile in Microsoft Visual Studio 2022.



5. The folder screen after compilation is completed



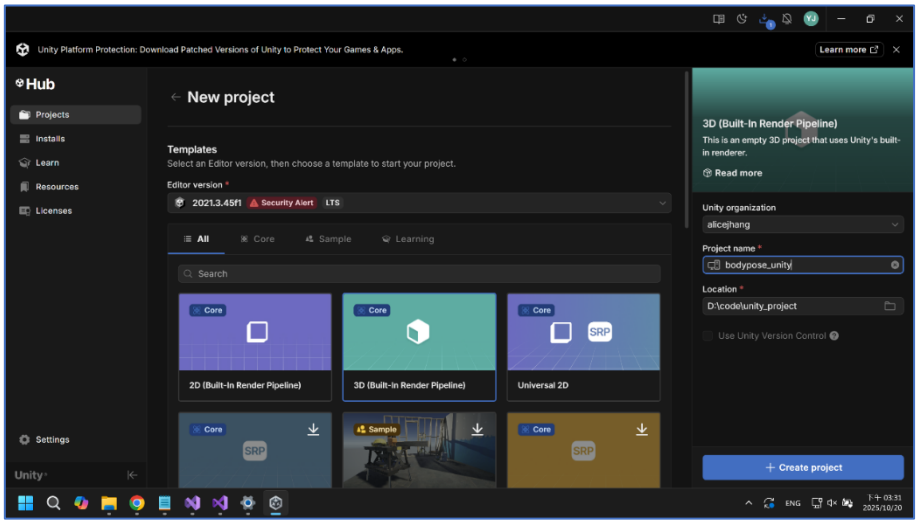
6. After modifying the camera information
   in %LIPSBODYPOSE_SDK_ROOT%bin, you can start the sample executable file

15. Unity

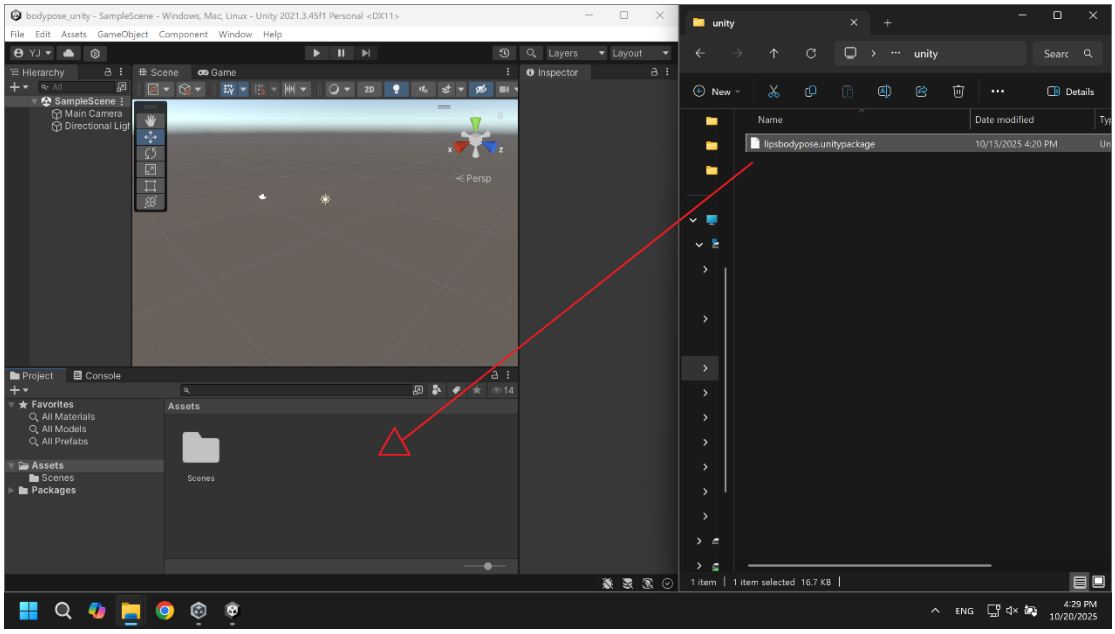
15.1 Unity Configuration Tutorial

This Unity integration empowers developers to deploy real-time, hardware-accelerated body pose detection with seamless performance. Follow these instructions to ensure accurate installation of native binaries, proper backend configuration, and reliable runtime execution.

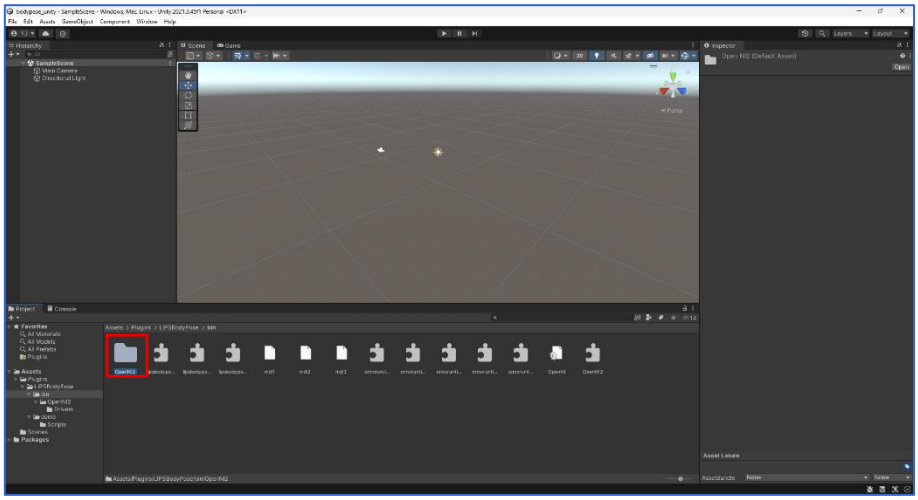
1. Create / open the Unity project.



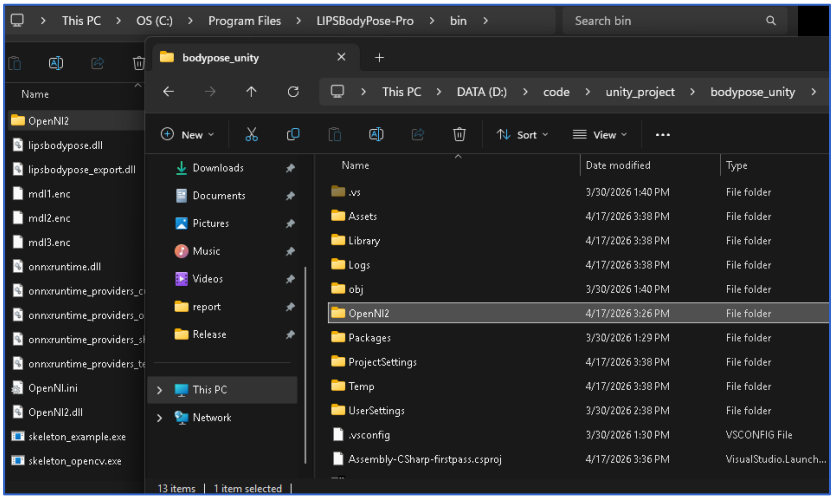
2. Import the **lipsbodypose.unitypackage** to the Unity project.
   - **Location**: C:\Program Files\[LIPSBodyPose]\unity



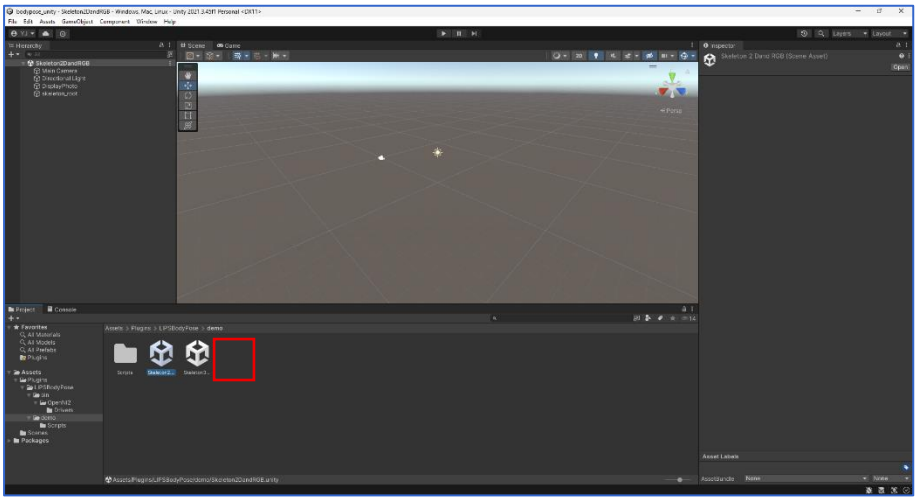
3. Modify OpenNI settings



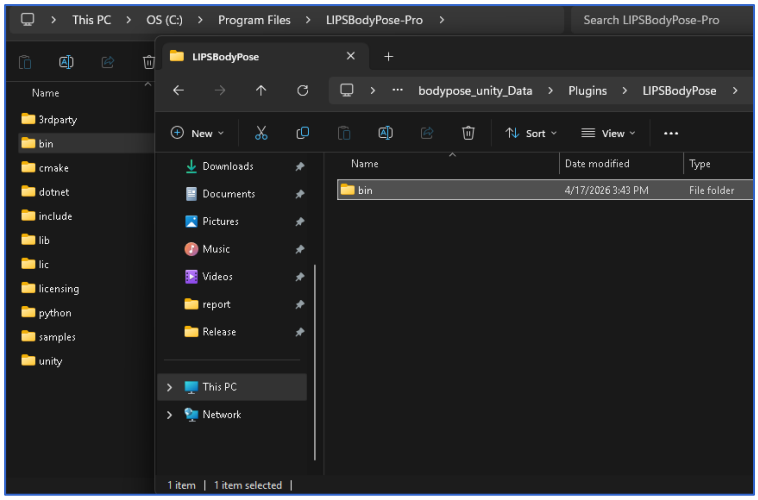
4. When using a DL camera, please copy the %LIPSBODYPOSE_SDK_ROOT%\bin\OpenNI2 folder to the project root directory.



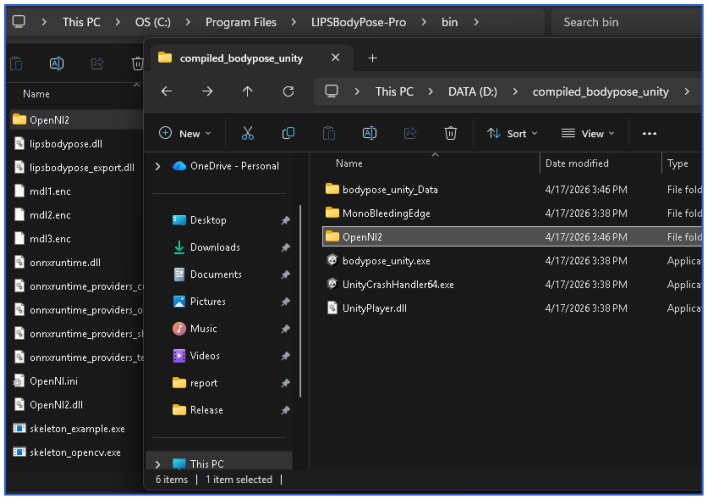
5. Run the Unity project. (To switch cameras for testing, please restart the Unity project first.)

6.  After compiling the Unity sample according to the Unity workflow, please copy the folder to the specified location.

   A. Create a new folder named ‘LIPSBodyPose’ and copy %LIPSBODYPOSE_SDK_ROOT%\bin into the LIPSBodyPose directory , then verify the contents of the OpenNI configuration file.



B. When using a DL camera, please copy the %LIPSBODYPOSE_SDK_ROOT%\bin\OpenNI2 folder to the directory containing the executable file.



LIPS CORPORATION

2F, No. 100, Ruiguag Road, Neihu District,
Taipei City 114, Taiwan
Tel.: + 886-2-8791-6998
Fax: +886-2-8791-8996
Official Website: https://www.lips-hci.com/
E-Mail: info@lips-hci.com