DepthVista

DepthVista RGB-D Mapping

Version 1.0

e-con Systems

1/22/2025

e-con Systems

Think Camera. Think e-con.

Disclaimer

e-con Systems reserves the right to edit/modify this document without any prior intimation of whatsoever.

Contents

INTRODUCTION TO DEPTHVISTA 4

DESCRIPTION 4

RGB-D MAPPING 5

RGB-D MAPPING 5
RGB-D MAPPING USING DEPTHVISTA SDK 5
POINT CLOUD 5

WORKFLOW 7

GENERATE 3D POINT CLOUD 9

INSTALLING DEPTHVISTASDK IN LINUX 10

INSTALLING DEPTHVISTASDK IN WINDOWS 11

PYTHON SCRIPT 12

REQUIREMENTS 12
LAUNCHING DEPTHVISTA PYTHON SCRIPT IN LINUX 12
LAUNCHING DEPTHVISTA PYTHON SCRIPT IN WINDOWS 12
SELECTING THE CAMERA DEVICE 12
CAPTURING FRAMES 13
EXITING APPLICATION 13

TROUBLESHOOTING 14

GLOSSARY 15

SUPPORT 16

Introduction to DepthVista

DepthVista is a 3D camera based on Time of Flight (TOF) technology, USB Video Class (UVC) compliant, USB 3.2 Gen 1 SuperSpeed USB camera from e-con Systems, which has over two decades of experience in designing, developing, and manufacturing OEM cameras.

DepthVista is a RGB-D camera containing both RGB and TOF depth cameras. RGB camera has 1/2.6" AR0234CS CMOS digital image sensor with global shutter from onsemi™. It has dedicated high performance color image signal processor. TOF depth camera has 1/4" CCD sensor and dedicated depth processor. DepthVista is a two-board solution containing camera board with the USB 3.2 Gen 1 interface and laser board along with enclosure.

This document describes the special features of sample camera application when it is used with DepthVista.

Description

DepthVista has USB interface controller with USB Type-C connector to interface with the host PC. It is a ready-to-manufacture camera board with all the necessary firmware built-in and is compatible with the UVC version 1.0 standard. You can integrate this camera into the products, and this helps to cut short the time-to-market.

DepthVista is a UVC compatible and will work with the standard drivers available with Windows and Linux OS. There is no need for any additional driver installation. So, video streaming through UVC is possible without any special drivers on OSes that have built-in support for UVC standards.

RGB-D Mapping

This section describes how to map RGB and Depth steam and how RGB-D Mapping works.

RGB-D Mapping

The **DepthVista RGB-D camera module** is a stereo camera system capable of streaming both RGB and Depth data simultaneously. Since these streams originate from two separate sensors, their fields of view (FOV) inherently differ. To align the FOV of the Depth and RGB sensors, the following parameters are utilized:

- **Camera Matrix**: Defines the intrinsic parameters of the camera, such as focal length and principal point.
- **Distortion Coefficients**: Corrects lens distortion in the captured images.
- **Transformation Matrix**: Applies a spatial transformation to align the Depth data with the RGB frame.
- **Rotation Matrix**: Handles rotational alignment between the two sensor perspectives.

These parameters are used to undistort and transform the Depth image, ensuring it matches the field of view of the RGB image. This alignment is essential for generating accurate and meaningful RGB-D data, particularly for applications like 3D point cloud creation and object detection.

RGB-D Mapping using DepthVista SDK

The **DepthVista SDK** provides an API called **SetRGBDMapping**, which allows enabling or disabling the alignment of RGB and Depth data. When enabled, this API applies the necessary transformations such as undistortion and register depth based on the calibration parameters, including the camera matrix, distortion coefficients, rotation matrix, and transformation matrix.

By enabling RGB-D mapping, the resolution and field of view (FOV) of the Depth and RGB streams are aligned.

Point Cloud

The DepthVista module comes with pre-calibrated **intrinsic** and **extrinsic camera matrices** for both **HD and VGA** resolutions, which are essential for accurate 3D construction.

Intrinsic Matrix contains the focal length and optical centre of the camera. Extrinsic Matrix defines the spatial relationship between the RGB and Depth sensors. These calibration

parameters can be retrieved using the **GetDeviceCalibrationParams** API provided in the DepthVista SDK.

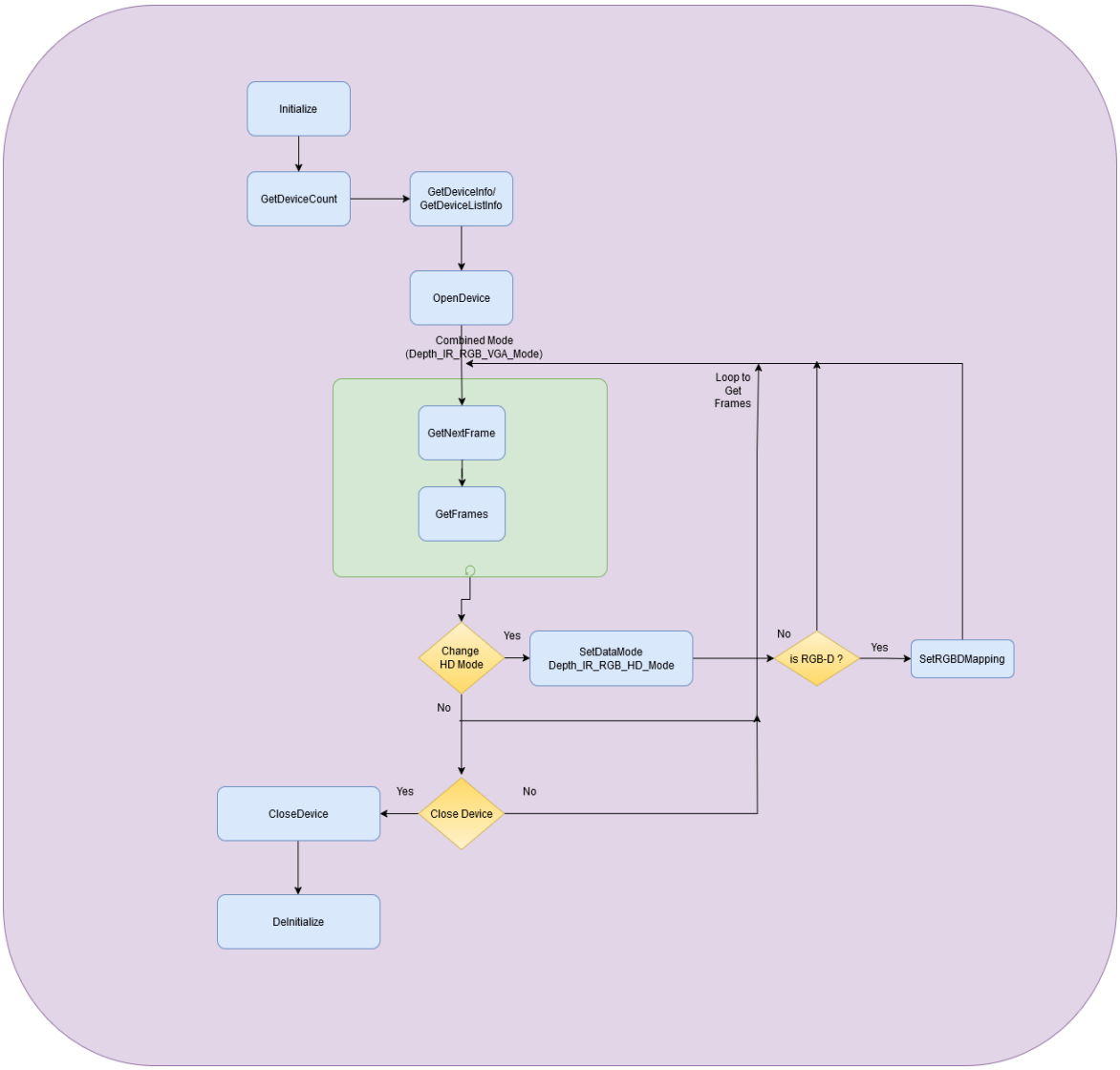
After aligning the Depth and RGB frames using the **SetRGBDMapping** API, a 3D point cloud can be generated by mapping the color data from the RGB frame onto the depth points. The intrinsic parameters of the RGB sensor are used during this process to ensure accurate projection of the color data onto the 3D geometry.

Workflow

This section describes the workflow of rgbdMapping.py script which can obtain RGB-D mapped stream using the **DepthVista SDK** and generate a 3D point cloud. The following steps outline the workflow implemented in the rgbdMapping.py script:

1. Initialize the DepthVista SDK
    - Use the **Initialize** API to initialize the DepthVista SDK and prepare it for operation.
2. Open the Device
    - Determine the number of DepthVista devices connected to the system using the **GetDeviceCount** API.
    - Retrieve device information using the **GetDeviceInfo** or **GetDeviceListInfo** API.
    - Use the retrieved device information to open the desired device with the **OpenDevice** API.
    - The **OpenDevice** API returns a **DeviceHandle**, which serves as an input parameter for subsequent API calls.
3. Set a DataMode that has support for both Depth and RGB streaming
    - Configure the data mode using the **SetDataMode** API. In this sample python script Depth_IR_RGB_HD mode is set. We can also use Depth_IR_RGB_VGA mode.
4. Enable RGB-D mapping
    - Enable RGB-D mapping using the **SetRGBDMapping** API.
5. Start a thread to Retrieve and Preview Frames
    - Use the **GetNextFrame** API to retrieve frames from the camera.
    - If **GetNextFrame** returns success, use the **GetFrames** API to get the processed and separated frames. We have introduced and used a structure called **Frames** which has all the separated frames such as rgb, ir, raw_depth and depth_colormap.
6. Obtain RGB-D Mapped Frames
    - After enabling RGB-D Mapping, GetFrames API will return mapped 2D depth frame and colormap frame. The resolution of depth_colormap frame and

raw_depth frame will match the resolution of RGB frame. Since we have set Depth_IR_RGB_HD mode in this sample, the resolution of RGB frame, depth_colormap frame and raw_depth frame will be 1280x720. If Depth_IR_RGB_VGA mode is used, the resolution of RGB frame, depth_colormap frame and raw_depth frame will be 640x480.



7. The script includes an option to capture the depth_colormap, raw_depth, RGB frames, and the 3D point cloud.

8. Generating 3D point cloud

    - To generate 3D point cloud, we need 2D raw_depth points, RGB frame and RGB camera's intrinsic paramters. Get RGB intrinsics and other calibration parameters using **GetDeviceCalibrationParams** API.

    The logic behind point cloud generation is discussed in detail below.

Generate 3D Point Cloud

This section describes how 3D point cloud is generated from 2D depth data.

The following data are vital for point cloud generation,

- Depth frame which has mapped 2D depth pixels (2D frame with depth data which is already mapped with RGB field of view)
- Corresponding RGB frame
- Focal length and principle point of RGB sensor. This can be found in the intrinsic matrix of the RGB sensor.

The following calculation converts the 2D depth coordinates (u, v) into 3D coordinates (x, y, z):

- Z = Depth[u, v] / zoom_factor
  where u is 2D x coordinate, v is 2D y coordinate and zoom_factor is a scaling constant. Here we consider zoom_factor as 1. So,
  Z = Depth[u, v]
  so Z is the depth value of a given pixel at u,v position.
- X = (u – principle point x) * Z / focal length x
  where principle point x is taken from the intrinsic matrix position [0,2]
  focal length x is taken from the Intrinsic matrix position [0, 0].
- Y = (v – principle point y) * Z / focal length y
  where principle point y is taken from the intrinsic matrix position [1,2]
  focal length y is taken from the Intrinsic matrix position [1, 1]

These points are collected to form a point cloud and saved as .ply file.

Installing DepthVistaSDK in Linux

This section describes the installation of DepthVistaSDK which is essential for building DepthVista Application.

The steps to install the DepthVistaSDK are as follows:

1. Run the following command to extract the **package** file.

```
Unzip -X <packageName.zip>
```

   <Extracted Directory>\linux\Bin\Ubuntu18.04\x64\SDK\DepthVistaSDKInstaller will have an install.sh file.

   Note: For Ubuntu 20.04 the install.sh file will be present in **<Extracted Directory>\linux\Bin\Ubuntu20.04\x64\SDK\DepthVistaSDKInstaller.**

2. Open the folder containing install.sh in terminal.

3. Run the following command to give executable permission for install.sh file.

```
chmod +x install.sh
```

4. Run the following command to install the **DepthVistaSDK**.

```
sudo ./install.sh
```

   Once installation is success, **Installation DepthVistaSDK success** message appears.

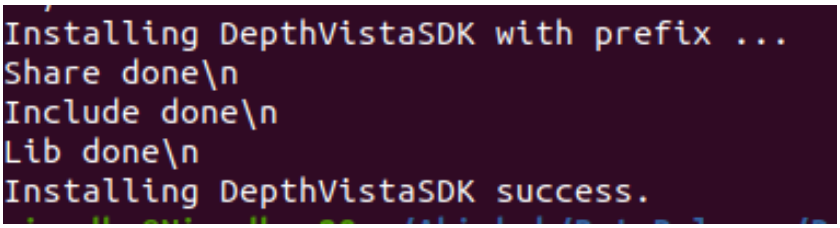


Fig 1: Installation success screenshot.

Installing DepthVistaSDK in Windows

The steps to install DepthVistaSDK in windows are as follows:

1. Extract the given package.

   <Extracted Directory>/Windows/Bin/SDK/Dll/x64 will contain the DepthVistaSDK.dll file.

   Note: Use x64 dll for x64 Python and x86 dll for X86 Python.

2. Copy the DepthVistaSDK.dll and opencv_world420.dll and place it in the Python Installation folder (root folder).

Python script

This section describes how to use the RGBDMapping Python Script.

Requirements

Install the below requirements to run the Python Script.

- Python version 3.6 or above
- OpenCV for python version 4.2 or above
- Numpy package
- DepthVista SDK

Launching DepthVista Python Script in Linux

The steps to launch RGBDMapping Python are as follows:

1. Open a terminal from the directory where rgbdMapping.py script is located
2. Run the following command to run the application.

```
sudo python3 rgbdMapping.py
```

Launching DepthVista Python Script in Windows

The steps to launch windows DepthVista Python are as follows:

1. Open command prompt from the directory where main.py script is located.
2. Run the following command to run the application.

```
python rgbdMapping.py
```

Selecting the Camera Device

Initially the command line displays the number of devices connected. You must select the camera device to explore their features.

```
                        e-con's Sample Python script for DepthVista
                    Demonstrates the working of e-con's DepthVistaSDK
                              DepthVista SDK-Version = 1.0.12

Number of Depth_vista (TOF) Devices connected: 1

        0.Exit
        1.See3CAM_TOF_25CUG     2F358B0029010900

Pick a Device to explore :      1
```

Figure 2: Application Launch Screen

Once the device is selected, stream will start with Depth_IR_RGB_HD streaming mode.

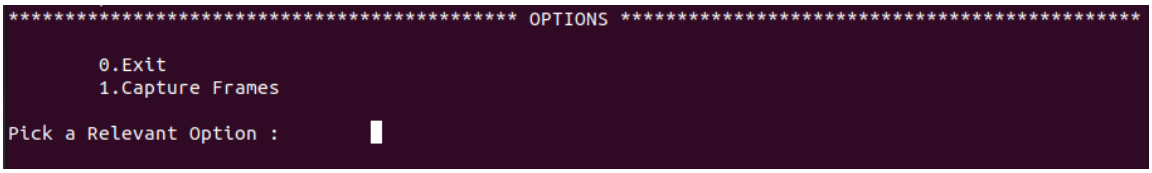


Figure 3: Selecting Options

Capturing Frames

Enter **2** in **Pick a Relevant Option** to capture images.

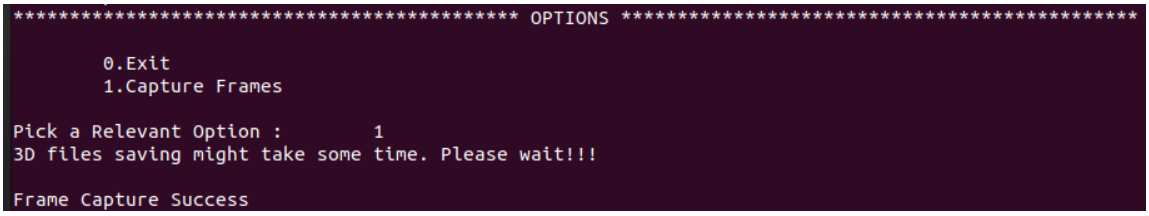


Figure 4: Capture Frames

The files will be saved with the name as shown below.

- RGB Frame – DepthVista_rgb_yyyy_mm_dd_hrs_min_sec.bmp
- Depth Raw Frame – DepthVista_Raw_yyyy_mm_dd_hrs_min_sec.raw
- Depth Color map - DepthVista_Depth_yyyy_mm_dd_hrs_min_sec.bmp
- 3D ply file – DepthVista_PLY_yyyy_mm_dd_hrs_min_sec.ply

Exiting Application

Enter **0** in **Pick a Relevant Option** to exit the application.

Troubleshooting

Error: error while loading shared libraries: libdc1394.so.25: cannot open shared object file: No such file or directory

Warning: libdc1394.so.25, needed by /usr/lib/libopencv_world.so.4.2.0, not found

Run the following command in terminal.

```
sudo apt-get install libdc1394-25
```

Glossary

FOV: Field Of View

USB: Universal Serial Bus

UVC Compliant: USB Video Class Compliant.

Support

Contact Us

If you need any support on DepthVista product, please contact us using the Live Chat option available on our website - https://www.e-consystems.com/

Creating a Ticket

If you need to create a ticket for any type of issue, please visit the ticketing page on our website - https://www.e-consystems.com/create-ticket.asp

RMA

To know about our Return Material Authorization (RMA) policy, please visit the RMA Policy page on our website - https://www.e-consystems.com/RMA-Policy.asp

General Product Warranty Terms

To know about our General Product Warranty Terms, please visit the General Warranty Terms page on our website - https://www.e-consystems.com/warranty.asp

Revision History

Rev	Date	Description	Author
1.0	22-January-2025	Initial Draft	Camera Products