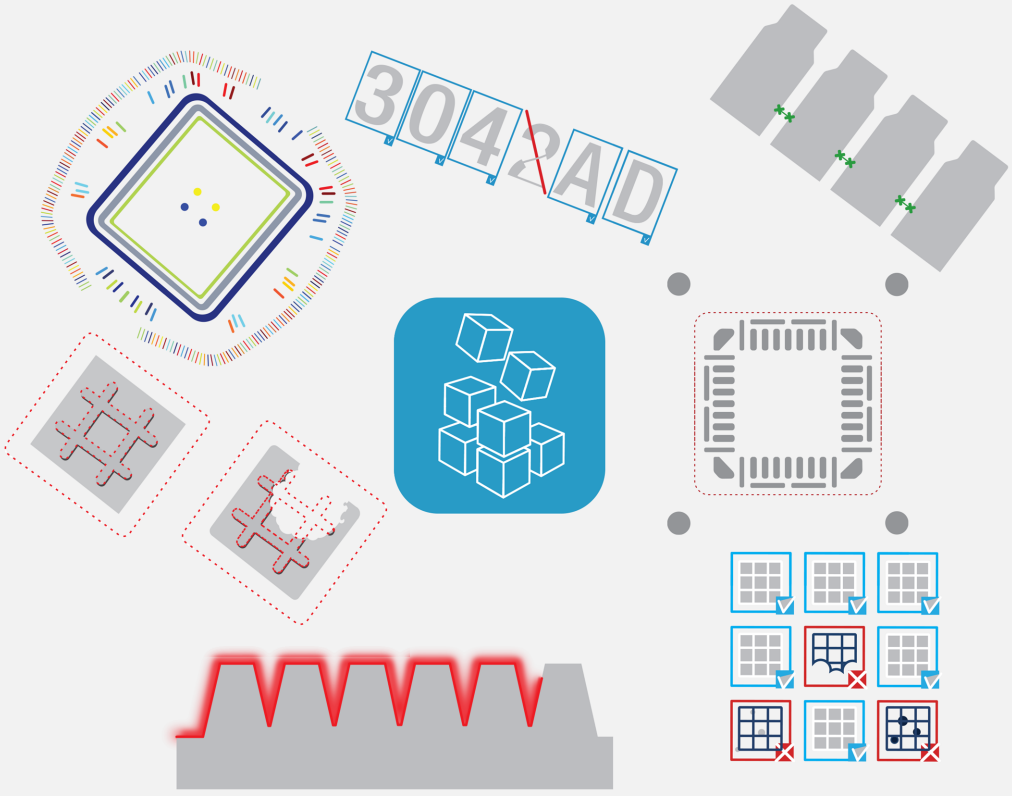
euresys
Empowering Computer Vision

USER GUIDE

Open eVision

Easy3D Compatibility with Shenzhen SinceVision 3D Sensors



This documentation is provided with Open eVision 2.15.0 (doc build 1147).
www.euresys.com

Easy3D Compatibility with Shenzhen SinceVision (SSZN) Technology 3D Sensors

Introduction

The **SSZN** 3D products are integrated laser triangulation sensors.

The specifications are available on the manufacturer website:
http://en.cnsszn.com/product/18/

- This document explains how to use the 3D data coming from these sensors with **Open eVision** 3D libraries and tools.
- A sample application distributed with source code demonstrates that integration. This application is freely available in the Easy3D Sensors Compatibility additional resources package on **Euresys** web site.

Resources

This document and the sample applications are based on the following resources:

- **SSZN** 3D sensor SR7050
- **EdgeImaging SDK** v3.3.2
- **Open eVision** 2.15
- Microsoft Visual Studio 2017

The **EdgeImaging SDK** is available on the manufacturer website:
http://en.cnsszn.com/download_list.html

- The C++ API is located in a folder named `SR7Link`.
- The **SSZNGrabberApp** project is configured with this folder and stored at:
  …\Easy3DGrab\SSZNGrabberApp\
- If you store it in another place, adapt your project configuration.

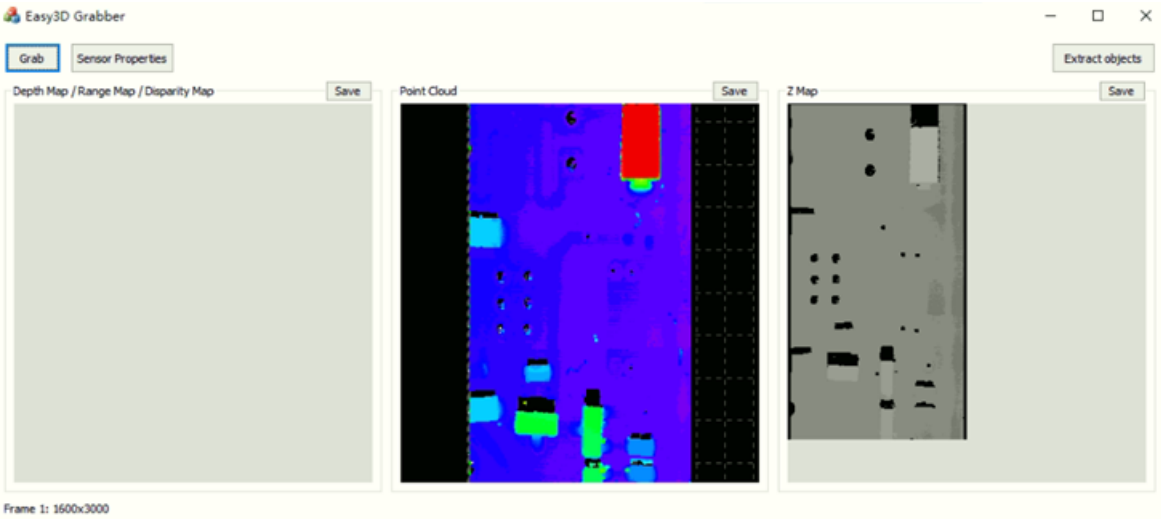
Features

- The **EdgeImaging SDK** provides a series of profile frames that consist of only Z-values. You can convert these profile frames a ZMap (`EZMap8` / `EZMap16` / `EZMap32f`).

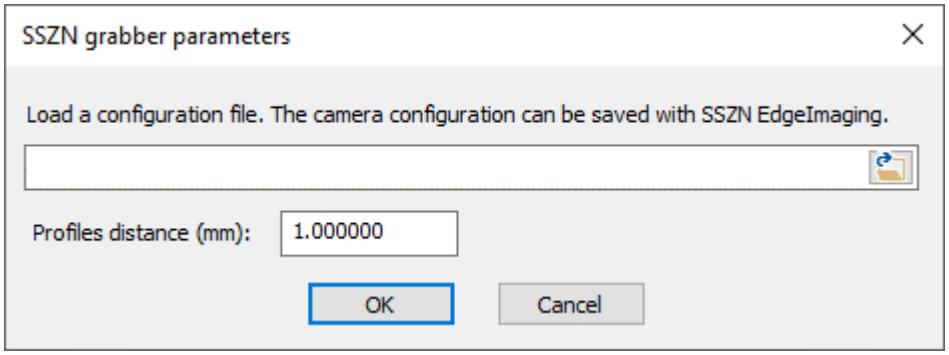
Easy3DGrab sample application

Easy3DGrab is distributed with C++ source code as an **Open eVision** additional resource.

- It features the acquisition of **SSZN** height data, the conversion to ZMaps and point clouds.
- You can save these representations.
- Click on the Grab button to acquire a new image.
- Open the Sensor Properties dialog to load a configuration file generated by **EdgeImaging** and configure the distance between 2 profiles (depending on the conveyor).
- The Object extraction function is exposed but you can use it only with the Easy3DObject license.



The Easy3DGrab application:
EDepthMap (left - not available), EPointCloud (center), EZMap (right)



The 3D sensor parameters: loading the configuration file and distance between 2 profiles

C++ code sample to convert the SSZN height data to Easy3D objects

Converting the SSZN height data to a ZMap

Here is the code snippet to fill an `Easy3D::EZMap16` object from the height data retrieved with the function `SR7IF_GetProfileData`:

```
int deviceID = 0;
SR7IF_Data dataObject = NULL;

if (SR7IF_ReceiveData(deviceID, dataObject) != 0)
{
  // Error
}

// number of profiles
int height = SR7IF_ProfilePointCount(deviceID, dataObject);

// profile width
int width = SR7IF_ProfileDataWidth(deviceID, dataObject);

int nbPoint = height * width;

// Retrieve the height data
int* heightData = new int[nbPoint];

if (SR7IF_GetProfileData(deviceID, dataObject, heightData) != 0)
{
  // Error
}

// Convert height data to ZMap
Easy3D::EZMap16 zmap;
zmap.SetSize(width, height);

// Set the resolution of the ZMap
float rx = float(SR7IF_ProfileData_XPitch(deviceID, dataObject)) / 1000.f;
float ry = profiles_distance / 1000.f;
float rz = 0.001f; // 1 mm per pixel
zmap.SetResolution(rx, ry, rz);

int i = 0;
for (int y = 0; y < height; ++y)
{
  uint16_t* dst = (uint16_t*)zmap.GetBufferPtr(0, y);

  // Copy height values to ZMap
  for (int x = 0; x < width; ++x, ++i)
  {
    dst[x] = uint16_t(heightData[i] / 100000); // transform value in mm
  }
}

delete[] heightData;
```

EPointCloud

- You cannot generate a point cloud directly from the **SSZN** 3D sensors.
- Generate a point cloud from the ZMap with the `Easy3D::EZMapToPointCloudConverter` class.

TIP
The sample application **Easy3DGrab** implement the `EPointCloud` and `EZMap` conversions.