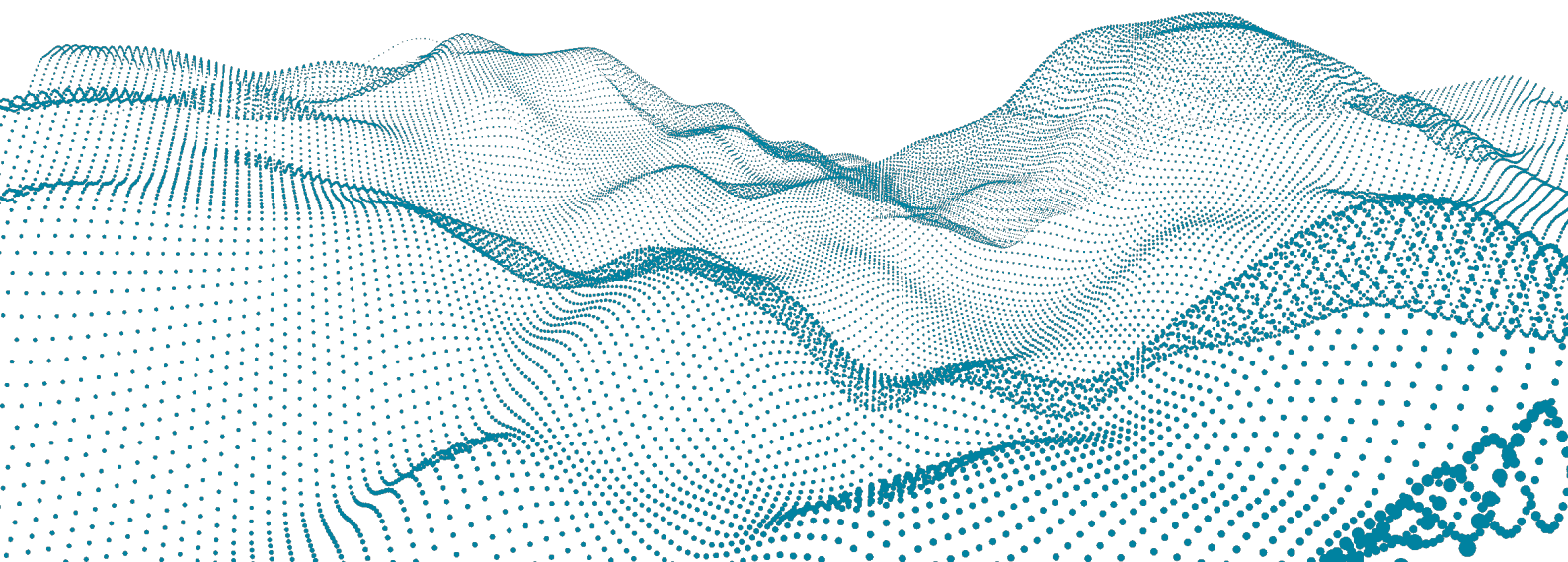


# roboception

Roboception GmbH | October 2022

## SGM<sup>®</sup> Producer

Release 22.10.0



# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Prerequisites . . . . .	2
<b>2</b>	<b>Installation</b>	<b>4</b>
2.1	Windows . . . . .	4
2.2	Ubuntu . . . . .	4
2.3	Other Linux Distributions . . . . .	4
2.4	Environment Variables . . . . .	5
<b>3</b>	<b>Tools for setup and calibration of the rc_viscore</b>	<b>6</b>
3.1	rc_check tool . . . . .	6
3.2	rc_calib tool . . . . .	6
<b>4</b>	<b>GenICam Nodemap</b>	<b>7</b>
<b>5</b>	<b>Using SGM<sup>®</sup> Producer in Applications</b>	<b>8</b>
5.1	Halcon . . . . .	8
5.2	C++ API . . . . .	8
5.3	OpenCV . . . . .	8
5.4	ROS . . . . .	8
<b>6</b>	<b>Troubleshooting</b>	<b>9</b>
<b>7</b>	<b>Appendix: Example Nodemap of an rc_viscore</b>	<b>11</b>

# 1 Introduction

The SGM<sup>®</sup>Producer is a software library for running stereo matching on a graphics card of a host computer for significantly increasing the frequency and reducing the latency for computing disparity images. It supports the Roboception *rc\_visard* and *rc\_viscore* stereo sensors. About 12.5 Hz can be reached for matching 1.2 MPixel images on a Nvidia GeForce RTX 2070. 4.6 Hz can be reached for matching 3 MPixel images on the same GPU. Higher frame rates are possible on more powerful Nvidia GPUs. Stereo matching without a graphics card on the CPU is also possible but significantly slower.

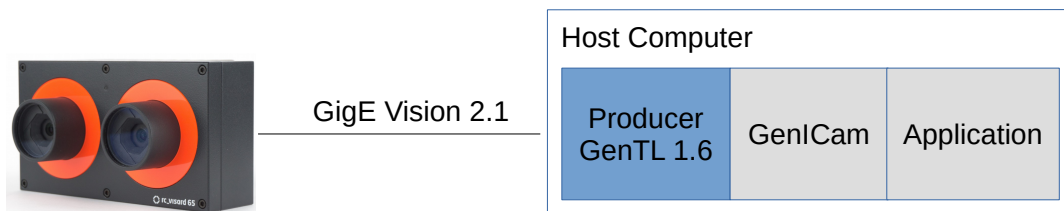


Figure 1: Overview of the *rc\_visard* and host computer. The producer is highlighted in blue.

In a GenICam application, the producer is responsible for communicating with the device, which in case of the *rc\_visard* uses the GigE Vision 2.1 standard.

The SGM<sup>®</sup>Producer from Roboception replaces a standard GigE Vision producer. It is compatible with all GenICam applications that use the GenTL 1.6 standard, e.g. Halcon from MVTec.

## 1.1 Prerequisites

The requirements on the host computer are:

- Supported operating systems include Windows 10 (64 bit), Ubuntu 18.04 (64 bit), Ubuntu 20.04 (64 bit), Ubuntu 22.04 (64 bit) and other 64 bit Linux distributions (producer only, without tools for checking and calibrating an *rc\_viscore*)
- On x86 architecture, the CPU must support the AVX2 instruction set. Almost all x86 CPUs that were manufactured from 2014 onward, especially the more powerful ones, provide this support.
- Nvidia GPU with compute capability 3.5 (i.e. Kepler architecture) or higher is highly recommended. The full functionality is also available without GPU support, but stereo matching will be very slow in this case.
- Supported sensor that is connected to the host computer.
- Application supporting GenICam

Supported sensors are:

- Roboception *rc\_visard* with firmware version 21.04.0 or higher
- Roboception *rc\_viscore*

The amount of required GPU memory and possible frame rates depend on the camera resolution, the configured depth quality, depth range and memory limitation using the `RC_SGM_MAXMEM` environment variable.

For *rc\_visard* sensors, a depth image frame rate of 25 Hz can be reached on mid-range Nvidia GPUs. The maximum amount of required GPU memory is below 1200 MB in all modes.

For *rc\_viscore* sensors, the following table shows the required GPU memory and resulting frame rates on an Nvidia GeForce RTX 2070 card for some example configurations. *rc\_viscore* sensors have an image resolution of 4112 x 3008 pixels and can deliver images with a maximum frame rate of 9 Hz.

DepthQuality	DepthMinDepth	DepthMaxDepth	RC_SGM_MAXMEM	GPU Memory	FPS
Low				553 MB	9 Hz
Medium				1057 MB	9 Hz
High				6043 MB	2.5 Hz
High	0.75 m			4343 MB	3.4 Hz
High	0.75 m	3 m	3312 MB	3639 MB	4.0 Hz
Full	1.6 m	3 m	7500 MB	7789 MB	1.8 Hz

Please note that stereo matching in full quality (i.e. 12 Mpixel) is in most situations not needed and not recommended. Instead, we recommend using the high quality setting with appropriate limitations on the depth range and maximum SGM memory as shown in the table above.

## 2 Installation

### 2.1 Windows

The software for Windows is provided as a ZIP archive that can be downloaded from <https://roboception.com/en/download/>.

The ZIP archive can be unpacked anywhere on the local hard drive. The main directory of the ZIP archive contains the file `rc_sgm_producer.cti`. This directory must be added to an environment variable named `GENICAM_GENTL64_PATH` so that applications can find the producer.

- Type “env” in the Windows search field (on a German Windows system type “Umgebung”)
- Select “Edit the system environment variables” (if you have administrator rights) or “Edit environment variables for your account”
- Check if `GENICAM_GENTL64_PATH` exists and create it as new if it does not exist
- Add the path to the directory in which `rc_sgm_producer.cti` is located to this environment variable
- Several paths can be specified with “;” as separation character
- Leave the dialog with “OK”

### 2.2 Ubuntu

The software for Ubuntu consists of the debian package `rc-sgm-producer*.deb`. It can be downloaded from <https://roboception.com/en/download/>. The package can be installed with:

```
sudo dpkg -i rc-sgm-producer*.deb
```

If there are errors due to missing packages, they can be installed by calling:

```
sudo apt install --fix-broken
```

The sub-directory `lib` contains the file `librc_sgm_producer.cti`, which must be included in the environment variable `GENICAM_GENTL64_PATH`. In the bash shell, this can be done by:

```
export GENICAM_GENTL64_PATH=$GENICAM_GENTL64_PATH:/opt/rc_sgm_producer/lib
```

The command line can be added to `.bashrc` for permanently including the path.

### 2.3 Other Linux Distributions

For other Linux distributions, the producer can be installed from the archive file `rc_sgm_producer*.tgz`. The producer in the TGZ package was built with minimal dependencies to support a wide range of Linux distributions. It can be downloaded from <https://roboception.com/en/download/>.

*NOTE: The TGZ package does not include the GUI tools for `rc_viscore`. Please use the Ubuntu or Windows packages for checking or calibrating an `rc_viscore`.*

The archive can be unpacked anywhere. It is recommended to unpack it under `/opt` as user root, e.g.

```
cd /opt
sudo tar xvfz <download-directory>/rc_sgm_producer*.tgz
```

The sub-directory `lib` contains the file `librc_sgm_producer.cti`, which must be included in the environment variable `GENICAM_GENTL64_PATH`. In the bash shell, this can be done by:

```
export GENICAM_GENTL64_PATH=$GENICAM_GENTL64_PATH:/opt/rc_sgm_producer/lib
```

The command line can be added to `.bashrc` for permanently including the path.

## 2.4 Environment Variables

The following environment variables influence the behavior of the producer:

- **RC\_SGM\_PRODUCER\_LEVEL:** This variable enables printing log information. By default, logging is done on stdout of the application that uses the producer. See **RC\_SGM\_PRODUCER\_LOG** for logging into a file. The possible log levels are given below. Higher levels include lower ones.
  - 0: Off (default)
  - 1: Fatal
  - 2: Error
  - 3: Warning
  - 4: Info
  - 5: Debug
  - 6: Trace (not recommended for general use as it can slow down the producer significantly)
- **RC\_SGM\_PRODUCER\_LOG:** This variable can be defined with the full path and name of the log file. Logging into this file is done according to the log level (see **RC\_SGM\_PRODUCER\_LEVEL**).
- **RC\_SGM\_MAXMEM:** Maximum amount of memory in MB that can be used as temporary memory for stereo matching. If a compatible GPU is available, this is memory on the GPU. The depth range will be reduced if this amount of memory is reached. The default is 0, which means unlimited.

## 3 Tools for setup and calibration of the rc\_viscore

The rc\_check and rc\_calib tools only support Roboception rc\_viscore devices. For rc\_visard devices, setup (i.e. installation) and calibration is offered through rcdiscover and the Web GUI, as described in the rc\_visard manual <https://doc.rc-visard.com>.

*NOTE: These tools are not contained in the TGZ package. Use the Ubuntu or Windows packages for getting these tools.*

### 3.1 rc\_check tool

The tool rc\_check can be used for checking the network connection and basic setup of an rc\_viscore device. In case of a misconfiguration, the tool also helps to reset the rc\_viscore to factory defaults. The usage is self explanatory.

### 3.2 rc\_calib tool

The tool rc\_calib can be used for checking and adjusting the focus and calibration of rc\_viscore devices. After starting the tool, select *File / Connect sensor ...*, choose the rc\_viscore from the list and specify the size of the calibration grid. The tool will then show the live images from the left and right cameras.

For checking and adjusting the focus, select *Adjust Focus* and follow the instructions in <https://doc.rc-viscore.com/en/installation.html#adjust-focus>.

For checking the calibration and recalibration, select the appropriate calibration option. The procedure is the same as described in the rc\_cube manual at [https://doc.rc-cube.com/latest/en/camera\\_calibration.html#verify-calibration](https://doc.rc-cube.com/latest/en/camera_calibration.html#verify-calibration).

**It is mandatory to always check the calibration after mounting the rc\_viscore, changing the focus or aperture.**

## 4 GenICam Nodemap

The producer offers the same nodemap (i.e. parameters) for rc\_visard and rc\_viscore devices, although some may be disabled, depending on device capabilities. Most of the parameters are standard GenICam features. Their meaning and usage is described in the GenICam Standard Features Naming Convention (SFNC), e.g. [https://www.emva.org/wp-content/uploads/GenICam\\_SFNC\\_v2\\_7.pdf](https://www.emva.org/wp-content/uploads/GenICam_SFNC_v2_7.pdf).

There are also some proprietary parameters, which are described in the rc\_visard documentation at <https://doc.rc-visard.com/latest/en/gigevision.html#custom-genicam-features-of-the-rc-xxx>.

The appendix shows the full nodemap of an rc\_viscore device as an example.



## 5 Using SGM<sup>®</sup>Producer in Applications

The producer is a software library that implements the GenTL 1.6 interface. A GenICam compatible application is required for using the producer.

### 5.1 Halcon

Halcon fully supports the `rc_visard`, `rc_viscore` and the off-board producer. Halcon can only find the producer if the directory of the producer is specified in the environment variable `GENICAM_GENTL64_PATH` (see installation above). The GenICamTL package that is provided by MVTec must be installed additionally to Halcon.

For using the SGM<sup>®</sup>Producer in `hdevelop`, GenICamTL should be specified as the first parameter in the `open_framegrabber()` call. The device is identified either by the user-defined name or the device ID.

A good starting point to work with Halcon is the `rc_visard` example program (i.e. `gigevision2_roboception_rcvisard_objectmodel3d.hdev`) that is delivered with the Halcon GenICamTL package. The following modifications are needed for using it with the SGM<sup>®</sup>Producer:

- Specify 'GenICamTL' instead of 'GigEVision2' for the `open_framegrabber` command (the environment variable `GENICAM_GENTL64_PATH` must be set as described above).
- Remove or uncomment the line that is setting the parameter 'GevStreamDeliverIncompleteBlocks'. This parameter is not available in the SGM<sup>®</sup>Producer as it never publishes incomplete buffers.
- For color sensors, set the pixel format to 'RGB8'. The SGM<sup>®</sup>Producer does not support the pixel format 'YCbCr411\_8'.

### 5.2 C++ API

C++ programmers can use the `rc_genicam_api` convenience layer from Roboception that can be downloaded from [https://github.com/roboception/rc\\_genicam\\_api](https://github.com/roboception/rc_genicam_api). The package offers a C++ interface to GenICam and the producer. It includes a standard GigE Vision producer for communication with the `rc_visard`. The standard producer can be replaced by the SGM<sup>®</sup>Producer by setting the environment variable `GENICAM_GENTL64_PATH` to the directory of the SGM<sup>®</sup>Producer (see installation above).

The package contains tools for getting and setting parameters and receiving images. The tools also serve as examples for demonstrating the use of the API.

### 5.3 OpenCV

A tutorial for getting started with the `rc_visard` and OpenCV is provided at [https://tutorials.roboception.de/rc\\_visard\\_general/opencv\\_example.html](https://tutorials.roboception.de/rc_visard_general/opencv_example.html). The OpenCV example is based on the `rc_genicam_api` (see C++ API above). The SGM<sup>®</sup>Producer is used by setting the environment variable `GENICAM_GENTL64_PATH` to the directory of the SGM<sup>®</sup>Producer (see installation above).

### 5.4 ROS

A ROS driver for Roboception sensors is available for ROS 1 and ROS 2 on the ROS build farm. If ROS is already installed, the driver can be installed with:

```
sudo apt install ros-${ROS_DISTRO}-rc-genicam-driver
```

The SGM<sup>®</sup>Producer is used by setting the environment variable `GENICAM_GENTL64_PATH` to the directory of the SGM<sup>®</sup>Producer as discussed in the installation section above.

More information about the drivers is given in the readmes of the ROS 1 and ROS 2 drivers, i.e. [https://github.com/roboception/rc\\_genicam\\_driver\\_ros](https://github.com/roboception/rc_genicam_driver_ros) and [https://github.com/roboception/rc\\_genicam\\_driver\\_ros2](https://github.com/roboception/rc_genicam_driver_ros2).

## 6 Troubleshooting

NOTE: The SGM<sup>®</sup>Producer will never forward disparity, confidence or error images that are processed on-board the rc\_visard to avoid confusion. A normal GigE Vision producer should be used for enabling on-board processing.

If the application does not find the producer:

- Check and correct the GENICAM\_GENTL64\_PATH variable. Restart the application after changing the variable.

If an rc\_visard device cannot be discovered:

- Check with the rcdiscover tool <https://github.com/roboception/rcdiscover> that the device is listed and marked as 'reachable'. You may also verify that the Web GUI can be reached by double-clicking on the device in the rcdiscover window.
- Under Windows, the firewall or an anti-virus package may block the communication to network devices. Try to temporarily disable the firewall or anti-virus package to see if this is the cause.

If an rc\_visard is discovered, but cannot be opened:

- Ensure that the firmware version of the rc\_visard is at least 21.04.0. This can be checked on the system page of the rc\_visard Web GUI. If the firmware version is too low, an update can be downloaded from <https://roboception.com/en/download/> and uploaded via the Web GUI of the rc\_visard.

If an rc\_viscore device cannot be discovered or cannot be opened:

- Check with the included rc\_check tool (see tools for the rc\_viscore above) that the device is listed and marked as 'ok'. Follow the hints that are provided by this tool.
- If rc\_check cannot find the rc\_viscore or any individual cameras of it, then the IP configuration might be wrong. By default, the cameras try to find a DHCP server. If they cannot find one, e.g. because the cameras are connected directly to a computer as recommended, they assign themselves a link local address. In this case, the host port must be configured accordingly, which is the default under Windows, but must be done manually under Linux. Alternatively, a temporary IP address can be manually assigned with the rcdiscover tool <https://github.com/roboception/rcdiscover>. After starting the tool, uncheck "Only rc\_... devices" to see the individual cameras. The entry "Reachable" indicates if the IP configuration is correct. After setting a temporary IP address and making the cameras reachable, the IP configuration can be made permanent with the command line tool gc\_config from the rc\_genicam\_api package [https://github.com/roboception/rc\\_genicam\\_api](https://github.com/roboception/rc_genicam_api).

If rc\_visard or rc\_viscore images cannot be received or if received buffers are incomplete:

- Try to connect the device directly (i.e. without switch) to the computer. In case of using a switch with an rc\_viscore, the switch must be connected with at least 2.5 Gbit to the host computer.
- In case of an rc\_visard, check on the system page of the Web GUI that the link speed is 1000 Mbit/s. Replace the network cable or check the network port settings on your host computer if the link speed is lower than 1000 Mbit/s.
- For avoiding incomplete buffers, the MTU should be set to 9000 if possible. Under Windows, this is often called jumbo frames. Increasing the receive buffer may also help. More information about network setup and optimization is provided in a tutorial at [https://tutorials.roboception.de/rc\\_visard\\_general/network\\_setup.html](https://tutorials.roboception.de/rc_visard_general/network_setup.html).
- Under Windows, the firewall or an anti-virus package may block the communication to the device. Try to temporarily disable the firewall or anti-virus package to see if this is the cause.

If camera images can be received, but disparity, confidence and error images are not delivered:

- The most likely cause for never receiving disparity images is insufficient GPU memory. Try to reduce the depth quality with the GenICam parameter 'DepthQuality'. Additionally or alternatively, the disparity range can be reduced by the GenICam parameters 'DepthMinDepth' and 'DepthMaxDepth'. It is also possible to specify the maximum amount of memory with the environment

variable `RC_SGM_MAXMEM` (see above). In this case, the producer increases the minimum depth as necessary in order to not exceed the given amount of memory.

If disparity, confidence and/or error images are delivered but the frequency is lower than expected:

- Check the display name of the producer (i.e. GenICam system) in your application. You may also install `rc_genicam_api` from [https://github.com/roboception/rc\\_genicam\\_api](https://github.com/roboception/rc_genicam_api) and call `'gc_info -l'` in the command line. It shows the display name in the first few lines of the output. Ensure that the display name reports your GPU. If not, then stereo matching is performed on the CPU, which is much slower than running it on the GPU. If an Nvidia GPU is installed on the computer, try updating the graphics card driver.
- Reducing the depth quality with the GenICam parameter 'DepthQuality' and/or reducing the disparity range with the GenICam parameters 'DepthMinDepth' and 'DepthMaxDepth' will increase the framerate.

Still having problems?

- Create a log file by setting the environment variable `RC_SGM_PRODUCER_LEVEL` to 5 and `RC_SGM_PRODUCER_LOG` to the full path and name of the logfile to be generated. The log file is a normal text file that can be checked for errors or sent to Roboception support.
- Contact Roboception support via [support@roboception.com](mailto:support@roboception.com). The support team may ask you for a log file (see above).

## 7 Appendix: Example Nodemap of an rc\_viscore

```

Category: DeviceControl (RO)
  Enumeration: DeviceType (RO) [Transmitter Receiver Transceiver Peripheral]: Transmitter
  Enumeration: DeviceScanType (RO) [Areascan Linescan Areascan3D Linescan3D]: Areascan3D
  String: DeviceVendorName (RO): Roboception GmbH
  String: DeviceModelName (RO): rc_viscore 210m-16-12M-H1
  String: DeviceManufacturerInfo (RO): Roboception GmbH
  String: DeviceVersion (RO): 22.04.0
  String: DeviceFirmwareVersion (RO): 22.04.0
  String: DeviceSerialNumber (RO): GX042737
  String: DeviceID (RO): GX042737
  String: DeviceUserID (RW): GX042737
  Integer: DeviceSFNCVersionMajor (RO) [-9223372036854775808, 9223372036854775807]: 2
  Integer: DeviceSFNCVersionMinor (RO) [-9223372036854775808, 9223372036854775807]: 5
  Integer: DeviceSFNCVersionSubMinor (RO) [-9223372036854775808, 9223372036854775807]: 0
  String: DeviceManifestPrimaryURL (RO): local:rc_sgm_producer_v1.zip;10000000;5728
  String: DeviceManifestSecondaryURL (RO):
  Enumeration: DeviceTLType (RO) [GigEVision CameraLink CameraLinkHS CoaXPress USB3Vision Custom]:
    GigEVision
  Integer: DeviceTLVersionMajor (RO) [0, 65535]: 2
  Integer: DeviceTLVersionMinor (RO) [0, 65535]: 1
  Integer: DeviceTLVersionSubMinor (RO) [-9223372036854775808, 9223372036854775807]: 0
  Integer: DeviceLinkSelector (RO) [-9223372036854775808, 9223372036854775807]: 0
  Integer: DeviceLinkSpeed (RO) [-9223372036854775808, 9223372036854775807]: 12500000 Bps
  Integer: DeviceLinkConnectionCount (RO) [0, 4294967295]: 1
  Integer: DeviceStreamChannelCount (RO) [0, 4294967295]: 1
  Integer: DeviceStreamChannelSelector (RW) [0, 0]: 0
  Enumeration: DeviceStreamChannelType (RO) [Transmitter Receiver]: Transmitter
  Enumeration: DeviceStreamChannelEndianness (RO) [Big Little]: Little
  Integer: DeviceEventChannelCount (RO) [0, 4294967295]: 0
  Enumeration: DeviceCharacterSet (RO) [UTF8 ASCII]: UTF8
  Enumeration: DeviceRegistersEndianness (RO) [Big Little]: Big
  Command: TimestampLatch (WO)
  Integer: TimestampLatchValue (RO) [0, 9223372036854775807]: 0 ns
  Boolean: RcSystemReady (RO): 1
Category: ImageFormatControl (RO)
  Integer: WidthMax (RO) [0, 4294967295]: 4112
  Integer: HeightMax (RO) [0, 4294967295]: 6016
  Integer: Width (RO) [4, 4112]: 4112
  Integer: Height (RO) [4, 6016]: 3008
  Enumeration: PixelFormat (RW) [Mono8]: Mono8
  Enumeration: ComponentSelector (RW) [Intensity IntensityCombined Disparity Confidence Error
    RawCombined Calibration]: Intensity
  Boolean: ComponentEnable (RW): 1
  Integer: ComponentIDValue (RO) [-9223372036854775808, 9223372036854775807]: 1
  Integer: DecimationHorizontal (RO) [-9223372036854775808, 9223372036854775807]: 1
  Integer: DecimationVertical (RO) [-9223372036854775808, 9223372036854775807]: 1
Category: AcquisitionControl (RO)
  Enumeration: AcquisitionMode (RW) [SingleFrame Continuous MultiFrame]: Continuous
  Command: AcquisitionStart (RW)
  Command: AcquisitionStop (RW)
  Integer: AcquisitionFrameCount (RW) [1, 4294967295]: 0
  Float: AcquisitionFrameRate (RW) [1, 9]: 8 Hz
  Boolean: AcquisitionFrameRateEnable (RO): 1
  Float: ExposureTime (RW) [20, 2e+07]: 20000 us
  Enumeration: ExposureAuto (RW) [Off Continuous AdaptiveOut1 Out1High]: Continuous
  Float: ExposureTimeAutoMax (RW) [20, 2e+07]: 111111 us
  Integer: ExposureRegionWidth (RW) [0, 4112]: 0
  Integer: ExposureRegionHeight (RW) [0, 3008]: 0
  Integer: ExposureRegionOffsetX (RW) [0, 4112]: 0
  Integer: ExposureRegionOffsetY (RW) [0, 3008]: 0

```

```

Enumeration: AcquisitionAlternateFilter (RW) [Off OnlyHigh OnlyLow]: Off
Enumeration: AcquisitionMultiPartMode (RW) [SingleComponent SynchronizedComponents]:
  SingleComponent
  Float: RcExposureAutoAverageMax (RW) [0, 1]: 0.75
  Float: RcExposureAutoAverageMin (RW) [0, 1]: 0.25
Category: AnalogControl (RO)
  Enumeration: GainSelector (RO) [All]: All
  Float: Gain (RW) [0, 48]: 0 dB
  Float: Gamma (RW) [0.01, 10]: 1
Category: DigitalIOControl (RO)
  Enumeration: LineSelector (RW) [Out1 Out2 Out3 In1 In2 In3 In4]: Out1
  Enumeration: LineMode (RO) [Output]: Output
  Boolean: LineStatus (RO): 0
  Integer: LineStatusAll (RO) [0x0, 0xff]: 0x0
  Enumeration: LineSource (RW) [Low High ExposureActive ExposureAlternateActive]: Low
Category: RcCalibration (RO)
  Float: RcCalibrationGridWidth (RW) [0, 5]: 0 m
  Float: RcCalibrationGridHeight (RW) [0, 5]: 0 m
  Float: RcCalibrationSensitiveAreaScale (RW) [0.5, 10]: 1
  Enumeration: RcCalibrationState (RW) [Focus]: Focus
  Boolean: RcCalibrationAvailable (RO): 1
Category: FileAccessControl (RO)
  Enumeration: FileSelector (RW) [Calibration]: Calibration
  Enumeration: FileOperationSelector (RW) [Open Close Read Write Delete]: Open
  Enumeration: FileOpenMode (RW) [Read Write ReadWrite]: Read
  Command: FileOperationExecute (RW)
  Enumeration: FileOperationStatus (RO) [Success Failure Invalid]: Success
  Integer: FileOperationResult (RO) [0, 4294967295]: 0 B
  Integer: FileSize (RO) [0, 4294967295]: 0 B
  Integer: FileAccessOffset (RW) [0, 4294967295]: 0 B
  Integer: FileAccessLength (RW) [0, 4294967295]: 0 B
  Register: FileAccessBuffer (RW)
Category: Scan3dControl (RO)
  Enumeration: Scan3dDistanceUnit (RO) [Pixel]: Pixel
  Enumeration: Scan3dOutputMode (RO) [DisparityC]: DisparityC
  Float: Scan3dCoordinateScale (RO) [-1.79769e+308, 1.79769e+308]: 0.0625
  Float: Scan3dCoordinateOffset (RO) [-1.79769e+308, 1.79769e+308]: 0
  Boolean: Scan3dInvalidDataFlag (RO): 1
  Float: Scan3dInvalidDataValue (RO) [-1.79769e+308, 1.79769e+308]: 0
  Float: Scan3dFocalLength (RO) [-1.79769e+308, 1.79769e+308]: 4782.06 Pixel
  Float: Scan3dBaseline (RO) [-3.40282e+38, 3.40282e+38]: 0.210188 m
  Float: Scan3dPrincipalPointU (RO) [-1.79769e+308, 1.79769e+308]: 2056 Pixel
  Float: Scan3dPrincipalPointV (RO) [-1.79769e+308, 1.79769e+308]: 1504 Pixel
  Float: FocalLengthFactor (RO) [-3.40282e+38, 3.40282e+38]: 1.16295
  Float: Baseline (RO) [-3.40282e+38, 3.40282e+38]: 0.210188 m
Category: DepthControl (RO)
  Enumeration: DepthQuality (RW) [Low Medium High Full]: High
  Integer: DepthFill (RW) [0, 4]: 3 pixel
  Integer: DepthSeg (RW) [0, 4000]: 200 pixel
  Float: DepthMinConf (RW) [0, 1]: 0.5
  Float: DepthMinDepth (RW) [0.1, 100]: 0.1 m
  Float: DepthMaxDepth (RW) [0.1, 100]: 100 m
  Float: DepthMaxDepthErr (RW) [0.01, 100]: 100 m
  Enumeration: DepthAcquisitionMode (RW) [SingleFrame SingleFrameOut1 Continuous]: Continuous
  Command: DepthAcquisitionTrigger (RW)
  Boolean: DepthSmooth (RW): 1
  Boolean: DepthStaticScene (RW): 0
  Boolean: DepthDoubleShot (RW): 0
  Float: DepthExposureAdaptTimeout (RW) [0, 2]: 0 s
Category: ChunkDataControl (RO)
  Boolean: ChunkModeActive (RW): 0
  Enumeration: ChunkSelector (RW) [Image OffsetX OffsetY Width Height PixelFormat Timestamp
  LineStatusAll FrameID ComponentID ComponentIDValue Scan3dDistanceUnit Scan3dOutputMode

```

```

Scan3dCoordinateScale Scan3dCoordinateOffset Scan3dInvalidDataFlag Scan3dInvalidDataValue
Scan3dFocalLength Scan3dBaseline Scan3dPrincipalPointU Scan3dPrincipalPointV Components
PartIndex DecimationHorizontal DecimationVertical LineSource]: Image
Boolean: ChunkEnable (RO): 0
Integer: ChunkFrameID (NA)
Integer: ChunkTimestamp (NA)
Integer: ChunkLineStatusAll (NA)
Float: ChunkExposureTime (NA)
Float: ChunkGain (NA)
Integer: ChunkWidth (NA)
Integer: ChunkHeight (NA)
Integer: ChunkOffsetX (NA)
Integer: ChunkOffsetY (NA)
Enumeration: ChunkPixelFormat (NA)
Enumeration: ChunkComponentID (NA)
Integer: ChunkComponentIDValue (NA)
Enumeration: ChunkComponentSelector (NA)
Enumeration: ChunkScan3dDistanceUnit (NA)
Enumeration: ChunkScan3dOutputMode (NA)
Enumeration: ChunkScan3dCoordinateSelector (RW) [CoordinateC]: CoordinateC
Float: ChunkScan3dCoordinateScale (NA)
Float: ChunkScan3dCoordinateOffset (NA)
Boolean: ChunkScan3dInvalidDataFlag (NA)
Float: ChunkScan3dInvalidDataValue (NA)
Float: ChunkScan3dFocalLength (NA)
Float: ChunkScan3dBaseline (NA)
Float: ChunkScan3dPrincipalPointU (NA)
Float: ChunkScan3dPrincipalPointV (NA)
Integer: ChunkComponents (NA)
Integer: ChunkPartIndex (NA)
Integer: ChunkDecimationHorizontal (NA)
Integer: ChunkDecimationVertical (NA)
Enumeration: ChunkLineSelector (RW) [Out1 Out2 Out3 In1 In2 In3 In4]: Out1
Enumeration: ChunkLineSource (NA)
Boolean: ChunkLineStatus (NA)
Float: ChunkRcNoise (NA)
Float: ChunkRcOut1Reduction (NA)
Boolean: ChunkRcTest (NA)
Float: ChunkRcBrightness (NA)
Boolean: ChunkRcAutoExposureAdapting (NA)
Boolean: ChunkRcReducedDepthRange (NA)
Float: ChunkRcMinDepth (NA)
Float: ChunkRcMaxDepth (NA)
Float: ChunkRcCalibrationError (NA)
Category: TestControl (RO)
  Enumeration: TestPayloadFormatMode (RW) [Off MultiPart]: Off
Category: TransportLayerControl (RO)
  Integer: TLParamsLocked (RW) [-9223372036854775808, 9223372036854775807]: 0
  Integer: PayloadSize (RO) [0, 4294967295]: 173165044 B
Category: PtpControl (RO)
  Boolean: PtpEnable (RW): 0
  Command: PtpDataSetLatch (WO)
  Enumeration: PtpStatus (RO) [Initializing Faulty Disabled Listening PreMaster Master Passive
  Uncalibrated Slave]: Disabled
  Integer: PtpOffsetFromMaster (RO) [-2147483648, 2147483647]: 0 ns
Category: GigEVision (RO)
  Integer: GevStreamChannelSelector (RW) [0, 0]: 0
  Integer: GevSCPSPacketSize (RO) [0, 4294967295]: 7500
  Integer: GevSCPD (RW) [0, 1000]: 0 ns
  Integer: GevInterfaceSelector (RW) [0, 1]: 0
  Integer: GevMACAddress (RO) [0:0:0:0:0:0, ff:ff:ff:ff:ff:ff]: 0:c:8d:60:a6:f1
  Boolean: GevCurrentIPConfigurationDHCP (RW): 1
  Boolean: GevCurrentIPConfigurationPersistentIP (RW): 0

```

Boolean: GevCurrentIPConfigurationLLA (RO): 1  
Integer: GevCurrentIPAddress (RO) [0.0.0.0, 255.255.255.255]: 172.23.42.4  
Integer: GevCurrentSubnetMask (RO) [0.0.0.0, 255.255.255.255]: 255.255.255.240  
Integer: GevCurrentDefaultGateway (RO) [0.0.0.0, 255.255.255.255]: 172.23.42.1  
Integer: GevPersistentIPAddress (RW) [0.0.0.0, 255.255.255.255]: 172.23.42.4  
Integer: GevPersistentSubnetMask (RW) [0.0.0.0, 255.255.255.255]: 255.255.255.240  
Integer: GevPersistentDefaultGateway (RW) [0.0.0.0, 255.255.255.255]: 0.0.0.0