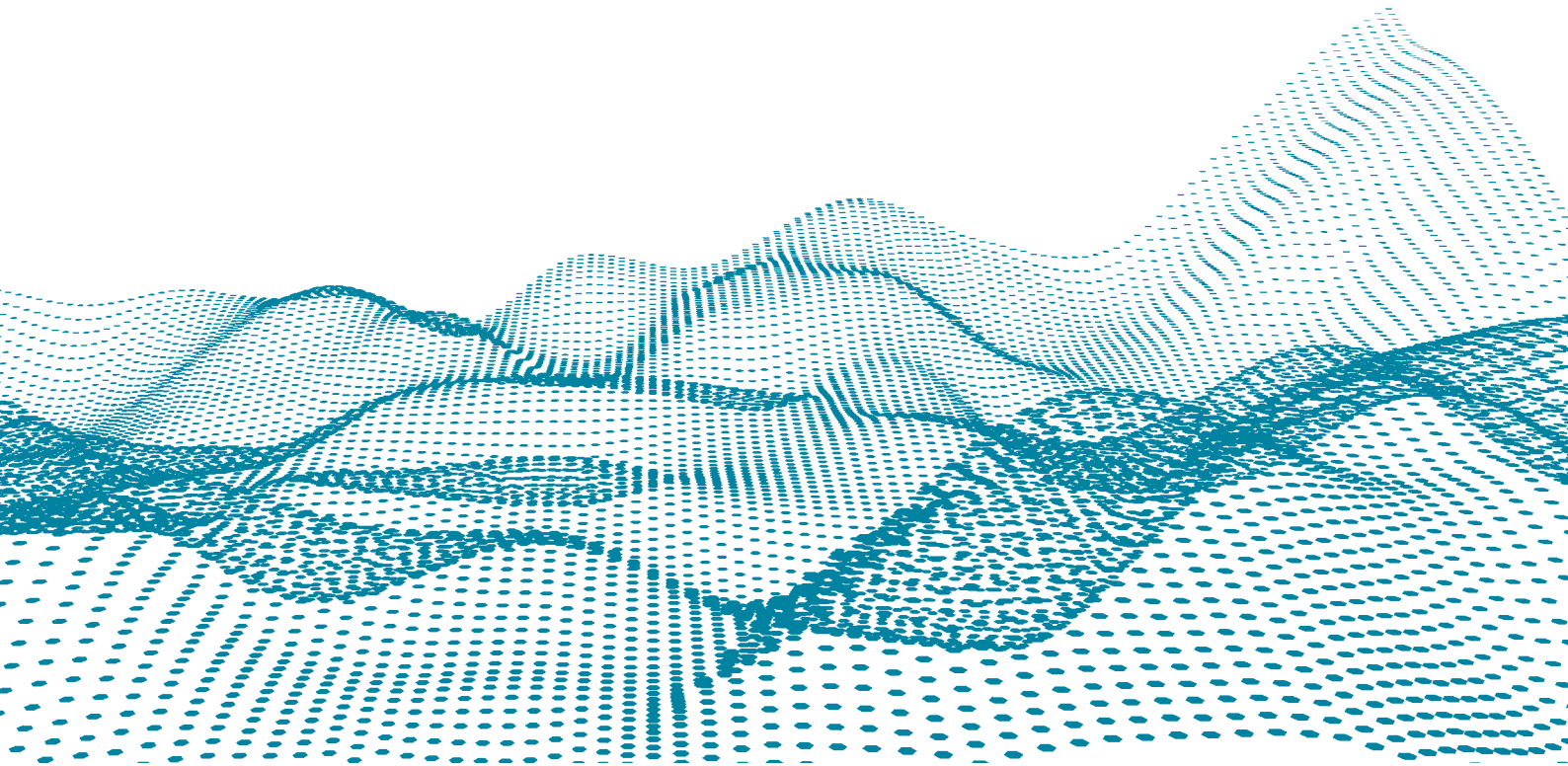


SGM® Producer
Release 21.07.0

Roboception GmbH

July 15, 2021



Contents

- 1 Introduction** **2**
- 1.1 Prerequisites 2
- 2 Installation** **3**
- 2.1 Windows 3
- 2.1.1 Installing 3
- 2.2 Ubuntu Linux 4
- 2.2.1 Installing 4
- 2.3 Environment Variables 4
- 3 Using SGM® Producer in Applications** **5**
- 3.1 Halcon 5
- 3.2 C++ API 5
- 3.3 OpenCV 5
- 3.4 ROS 5
- 4 Important Hints** **7**
- 5 Troubleshooting** **7**

1 Introduction

The SGM[®]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. About 12.5 Hz can be reached for matching 1.2 MPixel images on a Nvidia GeForce RTX 2070. Stereo matching without a graphics card on the CPU is also possible but significantly slower.

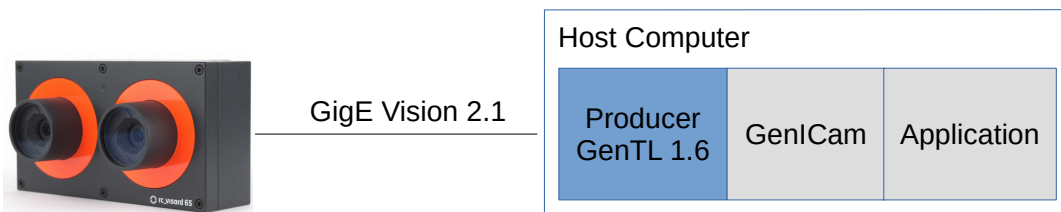


Figure 1: Overview of rc_visard and host computer. The producer is highlighted in blue.

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

The SGM[®]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:

- Operating system must be Windows 10 (64 bit), Ubuntu 20.04 or 18.04 (64 bit)
- Nvidia GPU with compute capability 3.5 (i.e. Kepler architecture) or higher with several GB free memory for each sensor that is connected and running in parallel. The amount of memory depends on the used quality and configured depth range.
- 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.

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/>.

2.1.1 Installing

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 Linux

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

2.2.1 Installing

Install the package by executing the following command in the directory where the debian package has been downloaded.

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

The directory that contains the file `librcsgmtl.cti` 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 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`).

3 Using SGM[®]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.

3.1 Halcon

Halcon fully supports the `rc_visard` 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). Additionally, the GenICamTL package that is provided by MVTec must be installed additionally to Halcon.

For using the SGM[®]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 of the `rc_visard` or the device ID that always starts with `rc_`.

A good starting point to work with the `rc_visard` in 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[®]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[®]Producer as it never publishes incomplete buffers.
- For color sensors, set the pixel format 'RGB8'. The SGM[®]Producer does not support the pixel format 'YCbCr411_8'.

3.2 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. 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[®]Producer by setting the environment variable `GENICAM_GENTL64_PATH` to the directory of the SGM[®]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.

3.3 OpenCV

A tutorial for getting started with `rc_visard` and OpenCV is provided at 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[®]Producer is used by setting the environment variable `GENICAM_GENTL64_PATH` to the directory of the SGM[®]Producer (see installation above).

3.4 ROS

A ROS driver is included 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
```

More information is given in the readmes of the ROS 1 and ROS 2 drivers, i.e. https://github.com/roboception/rc_genicam_driver_ros and https://github.com/roboception/rc_genicam_driver_ros2.

The SGM[®]Producer is used by setting the environment variable `GENICAM_GENTL64_PATH` to the directory of the SGM[®]Producer (see installation above).

4 Important Hints

Interface and device IDs always start with the prefix `rc_`. This permits to include the directory of a normal GigE Vision producer additionally to the SGM[®] Producer in the `GENICAM_GENTL64_PATH` variable. Accessing the device through an `rc_...` interface or with an `'rc_...'` device ID means to process stereo off-board.

The SGM[®] 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 using on-board processing.

5 Troubleshooting

If the application does not find the producer:

- Check and correct the `GENICAM_GENTL64_PATH` variable. Restart the application after changing the variable.

If the 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` 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 `rc_visard` images cannot be received or if received buffers are incomplete:

- Try to connect the `rc_visard` directly (i.e. without switch) to the computer.
- Check on the system page of the `rc_visard` Web GUI that the link speed is 1000. Replace the network cable or check the network port settings on your host computer if the link speed is lower than 1000.
- 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 optimisation is provided in a tutorial at 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 `rc_visard`. 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'.

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 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.
- Contact Roboception support via support@roboception.com. The support team may ask you for a log file (see above).