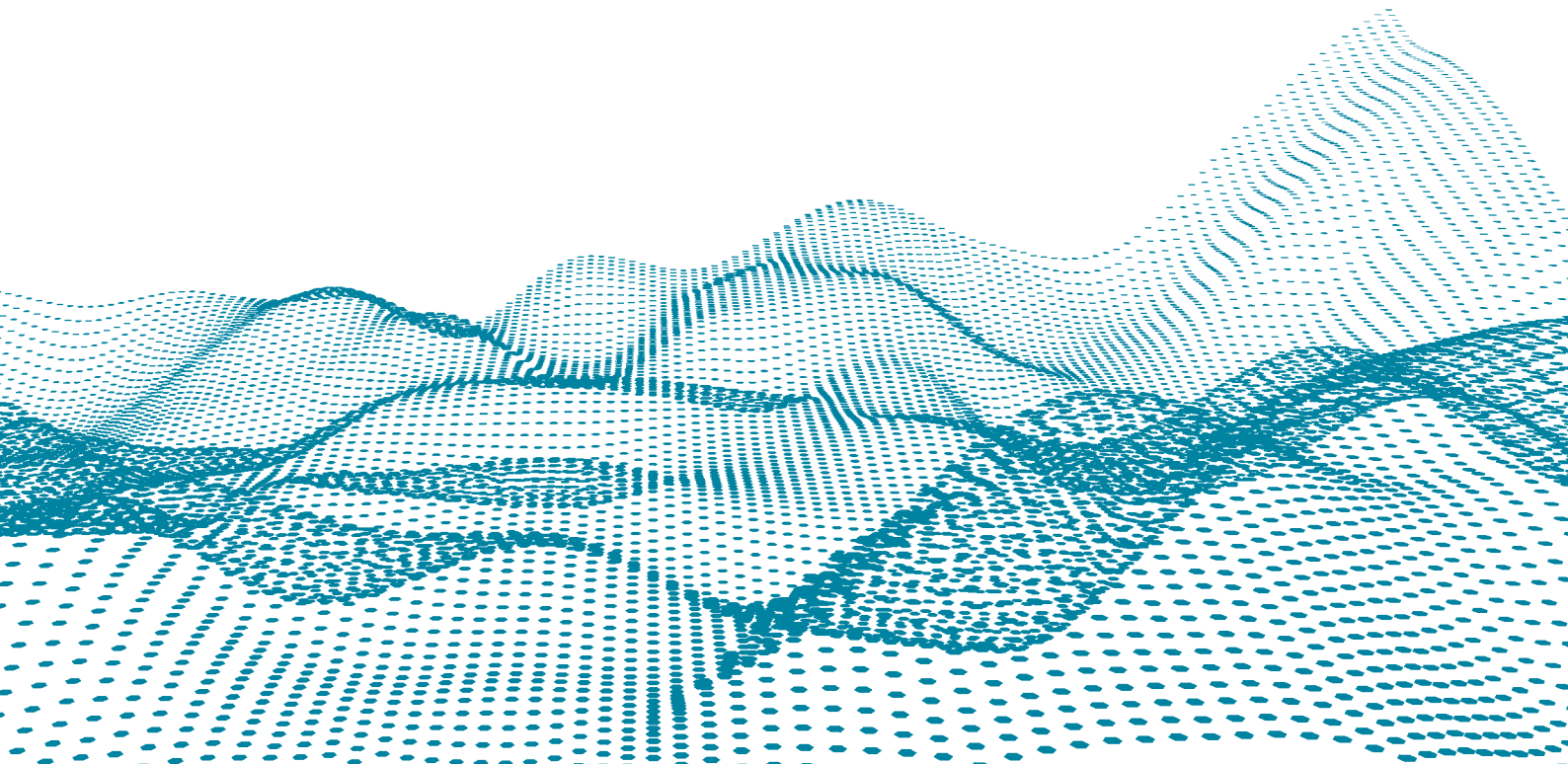


SGM[®] Producer
Release 21.04.0

Roboception GmbH

March 10, 2021



Contents

- 1 Introduction** **2**
- 1.1 Prerequisites 2
- 2 Installation** **3**
- 2.1 Windows 3
- 2.1.1 Installing 3
- 2.1.2 Testing 3
- 2.2 Ubuntu Linux 4
- 2.2.1 Installing 4
- 2.2.2 Testing 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** **6**
- 5 Troubleshooting** **6**

1 Introduction

The SGM[®]Producer is a software module 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 in full quality (i.e. matching 1.2 MPixel) on a Nvidia GeForce RTX 2070. Stereo matching without a graphics card on the CPU is also possible but significantly slower, although often faster than on-board processing, depending on the host CPU.

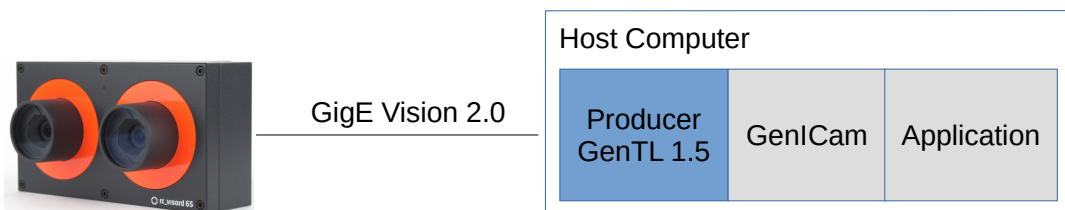


Figure 1: Overview of rc_visard and host computer. The producer is highlighted in blue.

In a GenICam application, the producer is responsible for communicating with the device, which in case of rc_visard uses the GigE Vision 2.0 standard.

The SGM[®]Producer from Roboception replaces a standard GigE Vision producer. Additionally to communicating to the rc_visard, it implements stereo matching for rc_visard devices on the host computer. It is compatible with all GenICam applications that use the current GenTL 1.5 standard, e.g. Halcon from MVTec. There is no change in application code needed for switching between on-board and off-board processing.

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 up to 3 GB free memory for each rc_visard that is connected and running in parallel. The amount of memory depends on the used quality and disparity range.
- Free Gigabit ethernet port for connecting the rc_visard
- Free USB port for connecting the license dongle from Roboception GmbH
- Application supporting GenICam

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. Thereafter, the directory that contains the file `rcsgmt1.cti` must be added to the environment variable named `GENICAM_GENTL64_PATH`, as Applications find the producer on the system through this environment variable:

- Search for "env"
- 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 `rcsgmt1.cti` is located to this environment variable
- Leave the dialog with "OK"

Finally, the license dongle from Roboception must be plugged into any free USB port.

2.1.2 Testing

The unpacked directory contains the file `rcsgmt1_info.exe` that can be executed by a double click. The tool opens a window that shows if all pre-requisites for using the producer are met. If an Nvidia GPU is installed on the system, but not detected, then the driver of the graphics card should be updated. Drivers can be found on the website of Nvidia. The SGM[®]Producer will also work without a supported graphics card, but at a performance.

2.2 Ubuntu Linux

The software for Ubuntu consists of two debian packages, i.e. `rcsgmtl_*.deb` and `rclm-dinkey_*.deb` that can be downloaded from <https://roboception.com/en/download/>.

2.2.1 Installing

Installing the provided debian packages is done by

```
sudo dpkg -i rcsgmtl_*.deb rclm-dinkey_*.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:/usr/lib/rcsgmtl
```

The command line can be added to `.bashrc` for permanently including the path. Finally, the license dongle from Roboception must be plugged into any free USB port.

2.2.2 Testing

The `rcsgmtl_*.deb` package contains the tool `rcsgmtl_info` that can be executed from the shell. It shows if all pre-requisites for using the producer are met. If an Nvidia GPU is installed on the system, but not detected, then the driver of the graphics card should be updated. The SGM[®]Producer will also work without a supported graphics card, but at a lower performance.

3 Using SGM[®] Producer in Applications

The producer is a software library that implements the GenTL 1.5 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 newest GenICamTL package that is provided by MVTec must be installed additionally to Halcon.

A good starting point to work with the `rc_visard` in Halcon is the `rc_visard` example program that is delivered with the GenICamTL package.

For using the off-board stereo 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 `rcsgm_`.

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 transport layer. 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/setting parameters and receiving images. The tools 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 in the ROS build farm. If ROS is already installed, the driver can be installed with:

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

The documentation of the driver and links to tutorials are given at http://wiki.ros.org/rc_visard_driver.

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

The SGM[®]Producer supports certain versions of the rc_visard firmware. The producer is updated with each new release of the rc_visard firmware to support new firmware versions. The `rcsgmt1_info` tool reports the supported rc_visard versions.

Interface and device IDs always start with the prefix `rcsgm_`. This permits to include the directory of a normal GigE Vision producer additionally to the off-board stereo producer in the `GENICAM_GENTL64_PATH` variable. Accessing the device as `rcsgm_...` means to process stereo off-board.

The off-board stereo 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 producer does not deliver any images:

- Call the `rcsgmt1_info` tool as explained above and see if it shows any problems.
- 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.
- The network connection can also give problems. Try to connect the rc_visard directly to the computer, without any switches in between. The MTU should be set to 9000 if possible. Under Windows, this is often called jumbo frames. Increasing the receive buffer may also help.

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

- Call the `rcsgmt1_info` tool as explained above and see if it shows any problems.

If disparity images are delivered but the frequency is lower than expected:

- Call the `rcsgmt1_info` tool as explained above and see if the GPU is detected. Update the graphics card driver if it is not detected.
- Open the Web GUI of the rc_visard and check the frequency setting on the camera page.

For any other problem, contact the Roboception support: support@roboception.com