SceneScan / SceneScan Pro

User Manual

(v1.9) July 1, 2020

nerian

VISION TECHNOLOGIES

Contents

1   Functionality Overview

SceneScan and SceneScan Pro (both referred as "SceneScan" in this document) are embedded image processing systems for real-time stereo matching. SceneScan connects to a dedicated stereo camera or two industrial USB cameras, which are mounted at slightly different viewing positions. By correlating the image data from both cameras, SceneScan can infer the depth of the observed scene. The computed depth map is transmitted through gigabit ethernet to a connected computer or another embedded system.

In combination with the cameras, SceneScan is a complete 3D sensor system. In contrast to conventional solutions, however, SceneScan works passively. This means that no light needs to be emitted for performing measurements. This makes SceneScan particularly robust towards the illumination conditions, and it facilitates long-range measurements, the use of multiple sensors with overlapping field of views, and a flexible reconfiguration of the system for different measurement ranges.

2   SceneScan / SceneScan Pro Differences

Two different models exist for the given image processing system: SceneScan and SceneScan Pro. Both models provide the same functionality, however, SceneScan Pro has significantly more computational power when compared to SceneScan. This means that SceneScan Pro can process a given input stereo image much faster than SceneScan.

Thanks to the additional processing power, SceneScan Pro is also capable of processing higher image resolutions, color images and larger disparity ranges. Due to these benefits, SceneScan Pro can achieve a higher measurement accuracy than SceneScan. Table 1 contains a brief comparison between SceneScan and SceneScan Pro. A detailed comparison of the achievable frame rates at different image resolutions and disparity ranges can be found in Section 4.4 on page 5.

3   Included Parts

The following parts should be included when ordering a new SceneScan system from Nerian Vision Technologies:

- SceneScan / SceneScan Pro processing system

- 12 V DC power supply with interchangeable mains connectors for Europe, North America, United Kingdom and Australia

- Printed user manual

- Calibration board

Table 1: Comparison between SceneScan and SceneScan Pro.

	SceneScan	SceneScan Pro
Max. grayscale resolution	0.6 MP (for 1:1 ratio) 0.5 MP (for 4:3 ratio)	6.2 MP (for 1:1 ratio) 5.3 MP (for 4:3 ratio)
Max. color resolution	n/a	2.0 MP
Max. disparity	128 pixels	256 pixels
Max. frame rate	45 fps	100 fps
Supported image types	monochrome	monochrome / color

- Ethernet cable, 3 m

If any of these items re missing, then please contact customer support.

4 General Specifications

4.1 Hardware Details

Power supply	11 - 14 V DC
Power consumption	Less than 10 W without supplying camera power Up to 20 W with supplying camera power
Dimensions	$104.5 \times 105.5 \times 45$ mm without mounting brackets $104.5 \times 130 \times 45$ mm with mounting brackets
Weight	400 g
I/O	$2\times$ USB 3.0 host, gigabit ethernet, GPIO
Max. USB power	900 mA
Conformity	CE, FCC, RoHS

4.2 Stereo Matching

Stereo algorithm	Variation of Semi-Global Matching (SGM)
Max grayscale resolution	SceneScan (1:1 ratio): 800×800 pixels SceneScan (4:3 ratio): 800×600 pixels SceneScan Pro (1:1 ratio): 2496×2496 pixels SceneScan Pro (4:3 ratio): 2659×2000 pixels
Max. color resolution	SceneScan: n/a SceneScan Pro (1:1 ratio): 1408×1408 pixels SceneScan Pro (4:3 ratio): 1632×1232 pixels
Supported pixel formats	Mono8, Mono12, Mono12p, Mono12Packed, RGB8[1] BayerGR8[1], BayerRG8[1], BayerGB8[1], BayerBG8[1]

---

[1]SceneScan Pro only.

Disparity range	SceneScan: 64 to 128 pixels
	SceneScan Pro: 96 to 256 pixels
	(32 pixels increment)
Frame rate	SceneScan: up to 45 fps
	SceneScan Pro: up to 100 fps
Sub-pixel resolution	4 bits (1/16 pixel)
Post-processing	Consistency check, uniqueness check, gap interpolation, noise reduction, speckle filtering

4.3   Image Rectification

Horizontal displacement	-39 to $+39$ pixels
Vertical displacement	-39 to $+39$ pixels
Interpolation	Bilinear

4.4   Achievable Frame Rates and Image Resolutions

The maximum frame rate that can be achieved depends on the image size and the configured disparity range. Table 2 provides a list of recommended configurations. This is only a subset of the available configuration space. Differing image resolutions and disparity ranges can be used to meet specific application requirements.

5   Mechanical Specifications

5.1   Dimensions

Figures 1a and 1b show SceneScan as seen from the front and from the side. The provided dimensions are measured in millimeters.

Table 2: Maximum frame rate by image resolution and disparity range for SceneScan and SceneScan Pro.

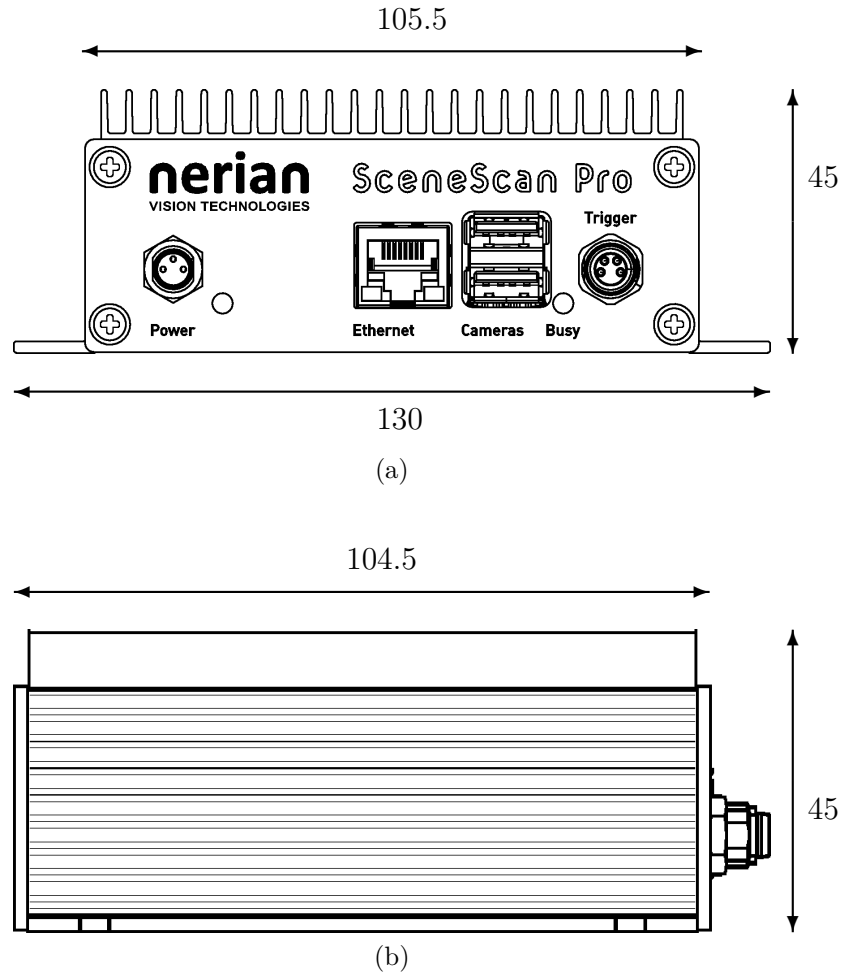| Model	Disparity Range	Image Resolution			
		640×480	800×592	1280×960	1600×1200
SceneScan	64 pixels	45 fps	30 fps	n/a	n/a
monochrome	128 pixels	30 fps	20 fps	n/a	n/a
SceneScan Pro	128 pixels	100 fps	65 fps	24 fps	15 fps
monochrome	256 pixels	70 fps	45 fps	15 fps	10 fps
SceneScan Pro color	256 pixels	45 fps	27 fps	9 fps	6 fps

Figure 1: (a) Front and (b) side view of SceneScan with dimensions in millimeters.
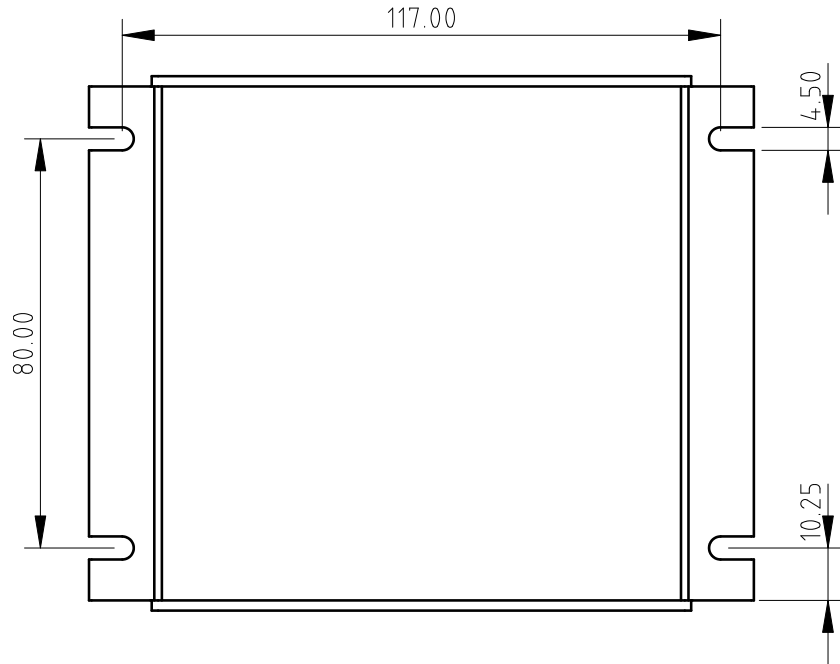
Figure 2: Dimensions of SceneScan mounting brackets.

5.2 Mounting

The casing of SceneScan features two mounting brackets to the sides of the device. Each mounting bracket has two slotted holes, which allows SceneScan to be mounted onto a flat surface. The dimensions and placement of the mounting brackets are depicted in Figure 2. In order to support heat dissipation, **it is recommended to mount the device onto a surface made of metal** or another material with a high heat conductivity.

6 Physical Interfaces

6.1 Interface Overview

Figures 3a and 3b show the interfaces on SceneScan's front and backside. These interfaces are:

Power connector: Connects to a power supply within the permitted voltage range.

Power LED: Indicates that the device is powered up and running.

Ethernet port: Port for connecting SceneScan to a client computer or another embedded system. This port is used for delivering processing results and for providing access to the configuration interface.
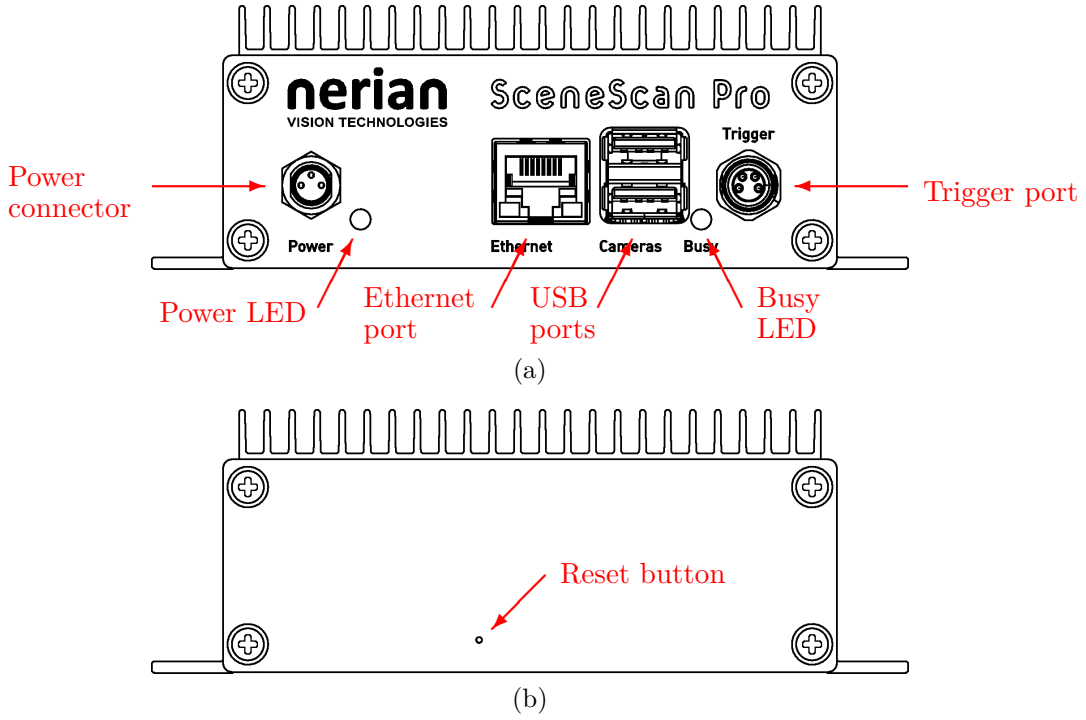
Figure 3: Interfaces on (a) front and (b) rear housing side.

USB ports: Ports for connecting SceneScan to up to two USB cameras. The maximum supply current of each port is 900 mA.

Busy LED: Indicates that the device is currently processing image data.

Trigger port: Provides a pulse signal for triggering both cameras. Also functions as an input for the time synchronization pulse.

Reset button: Button for resetting the device configuration back to the default state.

6.2 Power Supply

The power connector needs to be connected to the supplied power adapter or an equivalent model. When using an alternative power supply, please make sure that the voltage is in the permitted range of 11 - 14 V DC. Higher voltages might damage the device. A power supply with a maximum output current of at least 2 A is recommended, in order to supply sufficient power to the cameras (up to 900 mA on each USB port). If the cameras consume less than the permitted maximum current, or are powered externally, then a smaller power supply can be used.

The power connector uses a male 3 pin Binder 718/768 series connector. The pin assignment is shown in Figure 4. The following manufacturer part numbers correspond to matching connectors, and should be used for custom

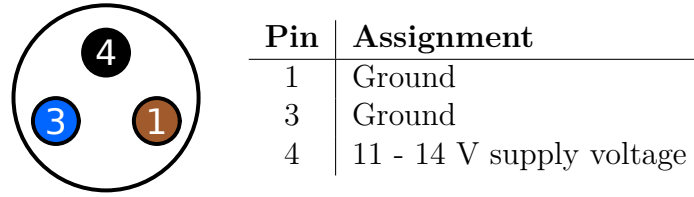| Pin	Assignment
1	Ground
3	Ground
4	11 - 14 V supply voltage

Figure 4: Pin assignment of power connector.

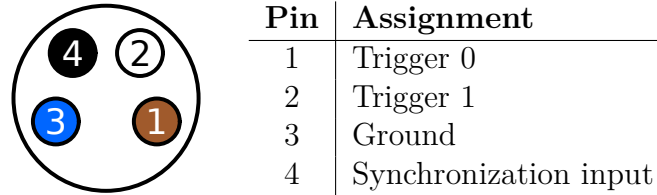| Pin	Assignment
1	Trigger 0
2	Trigger 1
3	Ground
4	Synchronization input

Figure 5: Pin assignment of trigger connector.

power supplies:

99 3400 00 03   Matching connector with solder termination.
99 3400 100 03   Matching connector with screw termination.
99 3400 550 03   Matching connector with cutting clamps termination.

6.3   Trigger Port

The trigger is a signal that is sent to the stereo camera pair. At the moment the signal is received, both cameras synchronously acquire an image. The trigger port can provide up to two +3.3 V pulse signals. The same port is also used for time synchronization through a synchronization pulse. The pin assignment is shown in Figure 5. Please refer to Sections 7.7 and 7.8 for further details on trigger and synchronization signals.

The trigger connector uses a female 4 pin Binder 718/768 series connector. The following manufacturer part numbers correspond to matching connectors, and should be used for custom trigger cables:

99 3383 00 04   Matching connector with solder termination, not shielded.
99 3383 100 04   Matching connector with screw termination, not shielded.
99 3383 500 04   Matching connector with cutting clamps termination, not shielded.
99 3363 00 04   Matching connector with solder termination, shielded.
99 3363 100 04   Matching connector with screw termination, shielded.

6.4   Configuration Reset Button

On the backside of the device is a hidden button that resets the stored configuration to the defaults. This button can be pressed by inserting a pin in the
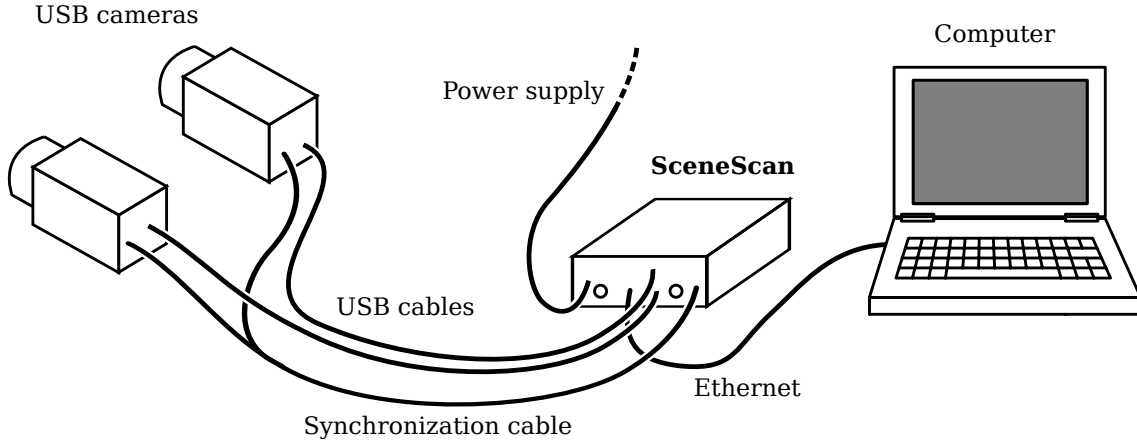
Figure 6: Example setup for cameras, SceneScan and client computer.

hole and pushing gently. The button needs to be pressed for at least 3 seconds immediately after the device is powered on. Please note that a configuration reset will also reset the network configuration. After a reset, the device will hence respond to the default IP address 192.168.10.10, unless a DHCP server in the network assigns a different address. A configuration reset should be performed if the device becomes unresponsive due to a misconfiguration.

7 Hardware Setup

7.1 Basic Setup

Figure 6 shows a basic system setup for stereo vision. A client computer that receives the computed depth data is connected to SceneScan's ethernet port. Two cameras are connected to the two available USB ports. As an alternative, it is possible to connect Nerian's dedicated Karmin2 and Karmin3 stereo cameras to one USB port.

The image acquisition of both cameras must be synchronized. SceneScan will only process frames with a matching time stamp. **Operating the cameras in free-run mode will thus result in dropped frames and erroneous results for non-static scenes**. In a typical configuration, synchronization requires connecting the cameras to SceneScan's trigger port (see Section 7.7), but it is also possible to use other external trigger sources.

7.2 Networking Configuration

It is recommended to connect SceneScan directly to the host computer's ethernet port, without any switches or hubs in between. This is because SceneScan produces very high-throughput network data, which might lead to packet loss when using network switches that cannot meet the required performance. It

must also be ensured that the host computer's network interface can handle an incoming data rate of up to 500 MBit/s.

For maximum performance, SceneScan should be configured to use Jumbo Frames (see Section 9.7). By default, Jumbo Frame support is disabled, as this requires an appropriate configuration of the host computer's network interface.

If SceneScan is accessible via the web interface and discovered in the devices list (e.g. in NVCom, see Section 11.1), but no image data is received (0 fps), this might indicate that Jumbo Frames are activated in SceneScan, but the network connection of the respective client computer is not properly configured to accept them.

In order to activate Jumbo Frame support in Windows, please follow the following steps:

1. Open 'Network and Sharing Center'

2. Open the properties dialog of the desired network connection

3. Press the button 'Configure...'

4. Open the 'Advanced' tab

5. Select 'Jumbo Packet' and choose the desired packet size (see Figure 7)

Please note that unlike Linux, Windows also counts the 14-byte ethernet header as part of the packet size. **When configuring SceneScan to use a 9000 bytes MTU, a Windows computer should thus be configured to use a 9014 bytes packet size.**

On Linux, Jumbo Frame support can be activated by setting a sufficiently large MTU, through the `ifconfig` command. For configuring a 9000 bytes MTU for interface eth0, please use the following command line:

```
> sudo ifconfig eth0 mtu 9000
```

Please be aware that the interface name might be different from eth0, especially in newer Linux releases.

The MTU is assigned automatically according to the SceneScan Jumbo Frame settings whenever a Linux computer receives configuration from an active SceneScan DHCP server (see Section 9.7). On Windows, automatic MTU assignment does not work, as Windows does not support this feature.

7.3 Supported Cameras

In addition to Nerian's own stereo cameras, Karmin2 and Karmin3, Scene-Scan and SceneScan Pro also support a variety of cameras from different vendors. SceneScan supports the USB3 Vision protocol, making it compatible to all cameras that correctly adhere to this standard. However, we recommend choosing a camera model for which compatibility has successfully been tested. Camera models with known compatibility are:
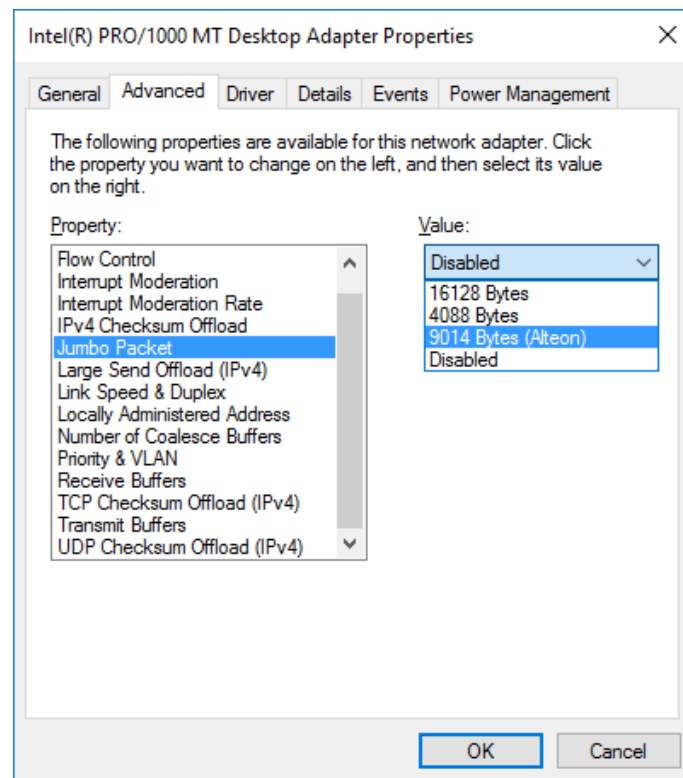
Figure 7: Jumbo Frames configuration in Windows

- Nerian Karmin2 stereo camera

- Nerian Karmin3 stereo camera

- Basler dart

- Basler ace

- FLIR Blackfly

- FLIR Blackfly S

- FLIR Chameleon3

- FLIR Flea3

- FLIR Grashopper3

- SVS-Vistek EXO

- The Imaging Source DMK

- Baumer CX series

- Baumer EX series

Figure 8: Example for valid stereo camera setup.

7.4 Color Camera Considerations

While SceneScan is limited to grayscale cameras, SceneScan Pro also supports color cameras. It should be noted that **results from grayscale cameras typically outperform the results from color cameras at equal resolutions**.

When using color cameras, SceneScan can process an RGB or Bayer pattern image. In the former case, the camera will need to perform Bayer pattern demosaicing. When SceneScan performs demosaicing it will at the same time downsample the image to half the original width and height. This is done in order to minimize artifacts that originate from Bayer demosaicing. Hence it is **strongly recommended to let SceneScan perform the Bayer demosaicing**, unless the full sensor resolution is required.

7.5 Camera Setup

Both cameras must have an exactly parallel viewing direction. They must be mounted on a plane with a displacement that is perpendicular to the cameras' optical axes. Furthermore, both cameras must be equipped with lenses that have identical focal lengths. **Because for most lenses adjusting the focus also slightly affects the focal length (known as lens breathing), the focus of both lenses should not be adjusted after the cameras have been calibrated**. An example for a valid camera setup is shown in Figure 8.

The distance between both cameras is referred to as *baseline distance.* Using a large baseline distance improves the depth resolution at higher distances. A small baseline distances, on the other hand, allows for the observation of close objects. The baseline distance should be adjusted in conjunction with the lenses' focal length. An tool for computing desirable combinations of baseline distance and focal length can be found online[2].

---

[2]http://nerian.com/support/resources/calculator/

7.6   Other Image Sources

SceneScan can also process image data that does not originate from real cameras. To allow for an easy evaluation, each device ships with an example stereo sequence on its internal memory. This example sequence appears as two *virtual cameras* that can be selected during camera configuration (see Section 9.4). If selected, the example sequence is replayed in an infinite loop. Due to speed limitations of the internal memory, the example sequence is not replayed at the full frame rate during the first loop iteration.

Another set of virtual cameras provide the ability to receive image data over the ethernet port. In this case a computer has to transmit a set of image pairs to SceneScan, and it then receives the processing results back over the same network. Please note that this significantly increases the required network bandwidth. The bandwidth usage can be reduced by transferring image data at a lower frame rate. **It is highly recommended to always use TCP as underlying network protocol when performing network transfer of input imagery**, in order to avoid dropped frames (see Section 9.7).

The NVCom client application that is described in Section 11.1 can be used for transferring a set of locally stored images to SceneScan. This can be achieved by pressing the appropriate button in the toolbar and selecting a directory with a collection of image files. The image files in this directory are then transmitted in alphabetical order. Please make sure that image files for the left camera always appear before their right camera counter parts when sorted alphabetically.

7.7   External Trigger

For stereo matching, it is important that both cameras are synchronized, meaning that both cameras record an image at exactly the same point of time. Many industrial cameras already feature the ability to synchronize themselves, by having one camera produce a trigger signal for the respective other camera.

As an alternative, SceneScan can produce up to two trigger signals. The signals are provided through the trigger port, which is described in Section 6.3. The peak voltage of both trigger signals is at $+3.3$ V and a maximum current of 24 mA can be supplied. The polarity of the trigger signals is active high.

For exact timing measurements, it is recommended that the cameras trigger on the rising signal edge. The pulse width and frequency can be adjusted in the trigger configuration (see Section 9.12).

If a new USB camera is plugged in, SceneScan will automatically activate the external trigger mode and select the first available trigger line. This might not match your trigger wiring. Hence, changing the camera's trigger settings might be required (see Section 9.5.1).

7.8   Time Synchronization Signal

As described in Section 6.3, one pin of the trigger port is dedicated to a time synchronization signal. If PPS time synchronization is activated in the device configuration (see Section 9.13), the internal clock is set to 0 whenever a rising signal edge is received on this pin. In order to trigger a synchronization, the signal must have a voltage level of at least 0.7 V. The maximum allowed voltage is 5.5 V.

Clock synchronization is useful when interpreting the timestamps that are embedded in the transmitted processing results (see Section 8.3). The synchronization input can be connected to the Pulse-Per-Second (PPS) output of a GPS receiver or a precision oscillator, in which case the clock is reset once per second. This allows for the reconstruction of high-precision timestamps on the computer receiving SceneScan's processing results.

As an alternative to synchronizing to an external signal, SceneScan can also perform a clock synchronization through the Network Time Protocol (NTP) or Precision Time Protocol (PTP), as described in Section 9.13.

8   Processing Results

8.1   Rectified Images

Even when carefully aligning both cameras, you are unlikely to receive images that match the expected result from an ideal camera geometry. The images are affected by various distortions that result from errors in the cameras' optics and mounting. Therefore, the first processing step that is performed is an image undistortion operation, which is known as *image rectification.*

Image rectification requires precise knowledge of the cameras' projective parameters. These can be determined through camera calibration. Please refer to Section 9.6 for a detailed explanation of the camera calibration procedure.

Figure 9a shows an example camera image, where the camera was pointed towards a calibration board. The edges of the board appear slightly bent, due to radial distortions caused by the camera's optics. Figure 9b shows the same image after image rectification. This time, all edges of the calibration board appear perfectly straight.

When performing stereo matching, SceneScan additionally outputs the rectified left camera image. This allows for a mapping of features in the visible image to structures in the determined scene depth and vice versa.

8.2   Disparity Maps

The stereo matching results are delivered in the form of a *disparity map* from the perspective of the left camera. The disparity map associates each pixel in the left camera image with a corresponding pixel in the right camera image.
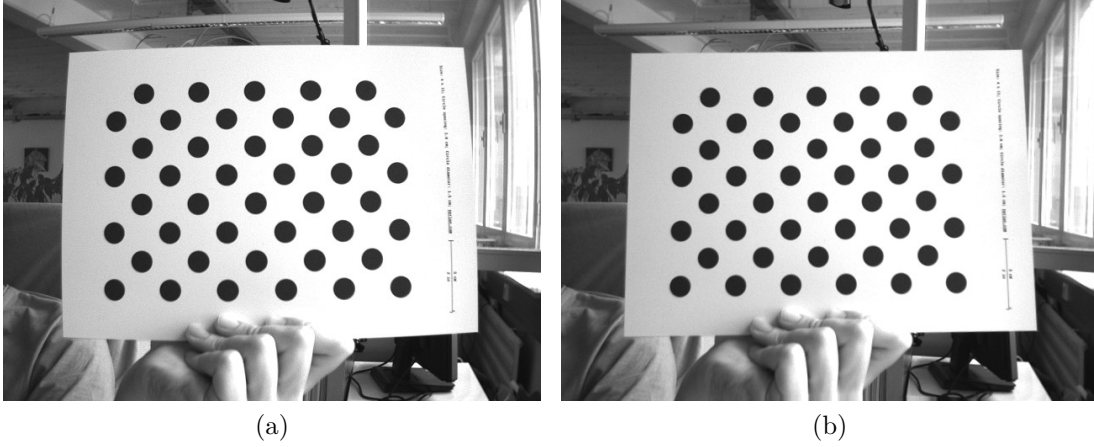
(a)   (b)

Figure 9: Example for (a) unrectified and (b) rectified camera image.

(a)   (b)
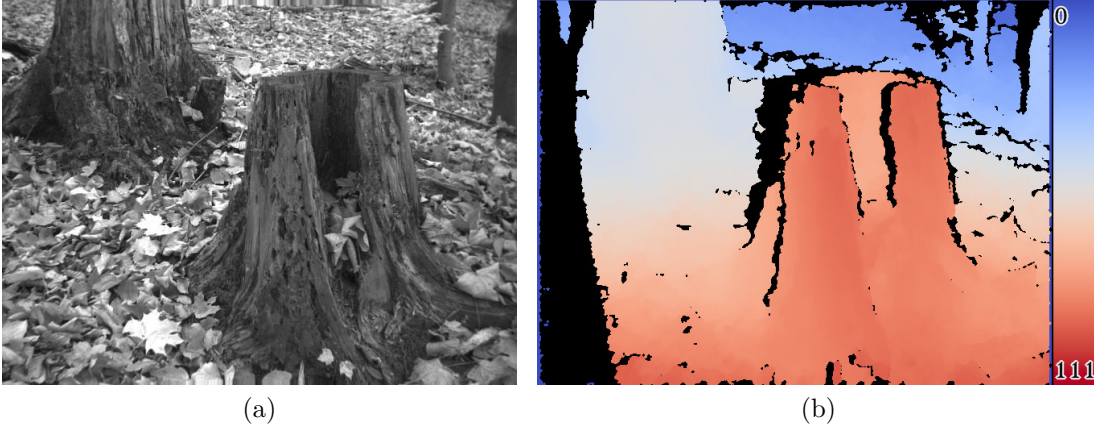
Figure 10: Example for (a) left camera image and corresponding disparity map.

Because both images were previously rectified to match an ideal stereo camera geometry, corresponding pixels should only differ in their horizontal coordinates. The disparity map thus only encodes a *horizontal coordinate difference.*

Examples for a left camera image and the corresponding disparity map are shown in Figures 10a and 10b. Here the disparity map has been color coded, with blue hues reflecting small disparities, and red hues reflecting large disparities. As can be seen, the disparity is proportional to the inverse depth of the corresponding scene point.

The *disparity range* specifies the image region that is searched for finding pixel correspondences. In the example image, the color legend indicates that the disparity range reaches from the minimum disparity $d_{min} = 0$ pixels to the maximum disparity $d_{max} = 111$ pixels. A large disparity range allows for very accurate measurements, but causes a high computational load, and thus lowers the achievable frame rate. SceneScan supports a configurable disparity

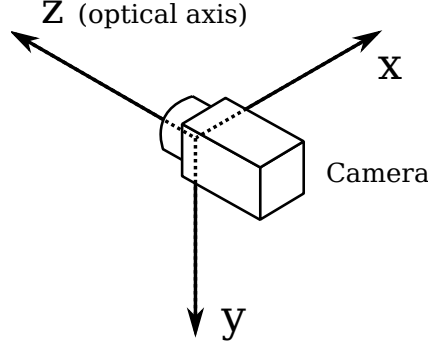Figure 11: Coordinate system used for 3D reconstruction.

range (see Section 9.9), which allows the user to choose between high-precision or high-speed measurements.

It is possible to transform the disparity map into a set of 3D points. This can be done at a correct metric scale if the cameras have been calibrated properly. The transformation of a disparity map into a set of 3D points requires knowledge of the disparity-to-depth mapping matrix Q, which is computed during camera calibration and transmitted by SceneScan along with each disparity map. The 3D location $\begin{pmatrix} x & y & z \end{pmatrix}^T$ of a point with image coordinates (u, v) and disparity d can be reconstructed as follows:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \frac{1}{w} \cdot \begin{pmatrix} x' \\ y' \\ z' \end{pmatrix}, \text{ with } \begin{pmatrix} x' \\ y' \\ z' \\ w \end{pmatrix} = Q \cdot \begin{pmatrix} u \\ v \\ d \\ 1 \end{pmatrix}$$

When using the Q matrix provided by SceneScan, the received coordinates will be measured in meters with respect to the coordinate system depicted in Figure 11. Here, the origin matches the left camera's center of projection. An efficient implementation of this transformation is provided with the available API (see Section 10.4).

SceneScan computes disparity maps with a disparity resolution that is below one pixel. Disparity maps have a bit-depth of 12 bits, with the lower 4 bits of each value representing the fractional disparity component. **It is thus necessary to divide each value in the disparity map by 16, in order to receive the correct disparity magnitude**.

SceneScan applies several post-processing techniques in order to improve the quality of the disparity maps. Some of these methods detect erroneous disparities and mark them as invalid. Invalid disparities are set to `0xFFF`, which is the highest value that can be stored in a 12-bit disparity map. In the example disparity map from Figure 10b, invalid disparities are depicted as black.

Please note that there is a stripe of invalid disparities on the left image border of the example disparity map. This behavior is expected as the disparity

map is computed from the perspective of the left camera. Image regions on the left edge of the left camera image cannot be observed by the right camera, and therefore no valid disparity can be computed. The farther left an object is located, the farther away it has to be, in order to also be visible to the right camera. The full depth range can only be observed for left image pixels with a horizontal image coordinate $u \geq d_{max}$.

8.3 Timestamps and Sequence Numbers

Each pair of rectified left camera image and disparity map, which is transmitted by SceneScan, also includes a timestamp and a sequence number. The timestamp is measured with microsecond accuracy and is set to either the time at which a camera trigger signal was generated or the time at which a frame was received from the cameras (see Section 9.12). For images that are received over the network, as described in Section 7.6, the timestamp and the sequence number are both copied.

As explained in Sections 7.8 and 9.13, it is possible to synchronize SceneScan's internal clock to an external signal or a time server. This directly affects the produced time stamps. When synchronized to a time server, time stamps are measured in microseconds since 1 January 1970, 00:00:00 UTC. If no synchronization is performed, the internal clock is set to 1 January 2017, 00:00:00 UTC at the time SceneScan is powered up. If synchronizing to an external signal, the clock is set to 0 at the rising edge of the synchronization signal.

Please note that synchronizing to a PPS signal, as explained in Section 7.8, also produces negative timestamps. Depending on the selected timestamp source, this happens when a synchronization signal is received while SceneScan is processing an already captured image pair, or while SceneScan is waiting for a frame corresponding to an already generated trigger signal. The negative timestamp is then the time difference between the reception of the synchronization signal and the time of capturing or triggering the current image pair.

9   Configuration

SceneScan is configured through a web interface, which can be reached by entering SceneScan's IP address into your browser. The default address is http://192.168.10.10, but if a DHCP server is present on the network, it might assign a different address to SceneScan. In this case please use the provided NVCom software for discovering the device (see Section 11.1). If SceneScan's integrated DHCP server is not enabled, please configure the IP address and subnet mask of your computer appropriately, such that this interface can be reached.

If SceneScan has just been plugged in, it will take several seconds before the web interface is accessible. **For using the web interface, you require**
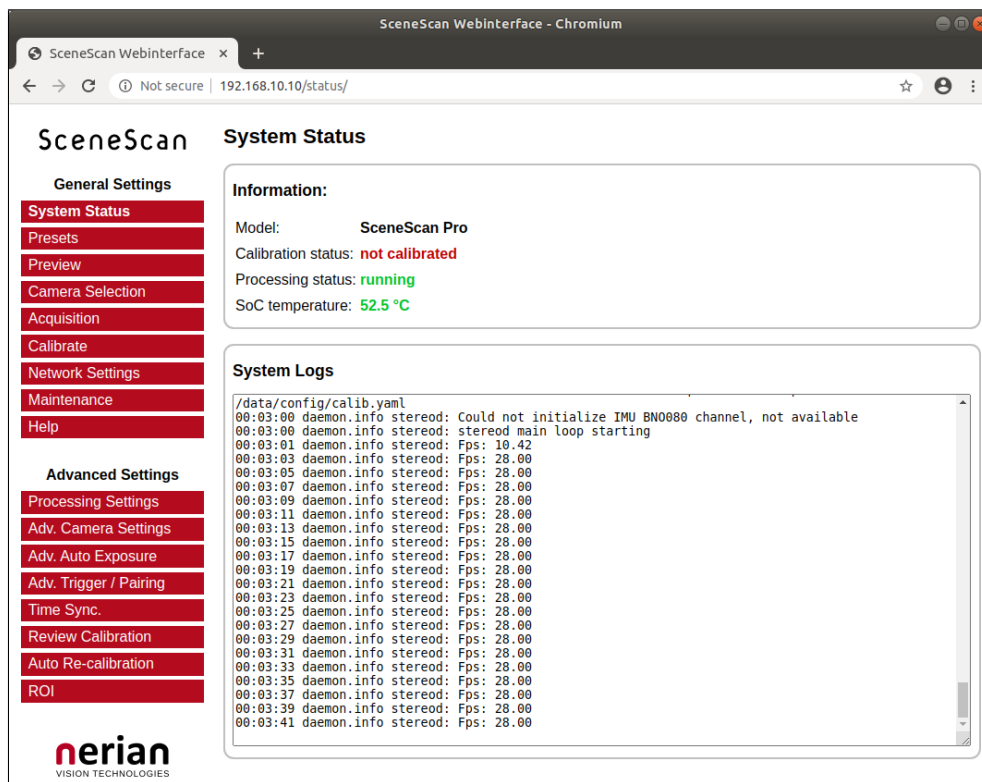
Figure 12: Screenshot of configuration status page.

a browser with support for HTML 5. Please use a recent version of one of the major browsers, such as Chrome, Firefox, Safari, or Edge.

The web-interface is divided into two sections: *General Settings* and *Advanced Settings*. The general settings pages contain the most commonly adjusted parameters. Modifying only these parameters should be sufficient for most applications. Less commonly adjusted parameters that might be relevant for very specific applications are found on the advanced settings pages.

9.1   System Status

The first page that you see when opening the web interface is the *'system status'* page that is shown in Figure 12. On this page, you can find the following information:

Model: Indicates whether the device is a SceneScan or SceneScan Pro model.

Calibration status: Provides information on whether the cameras have already been calibrated. Please be aware that some configuration changes will reset the camera calibration.

Processing status: Indicates whether the image processing sub-system has been started. If this is not the case, then there might be a problem
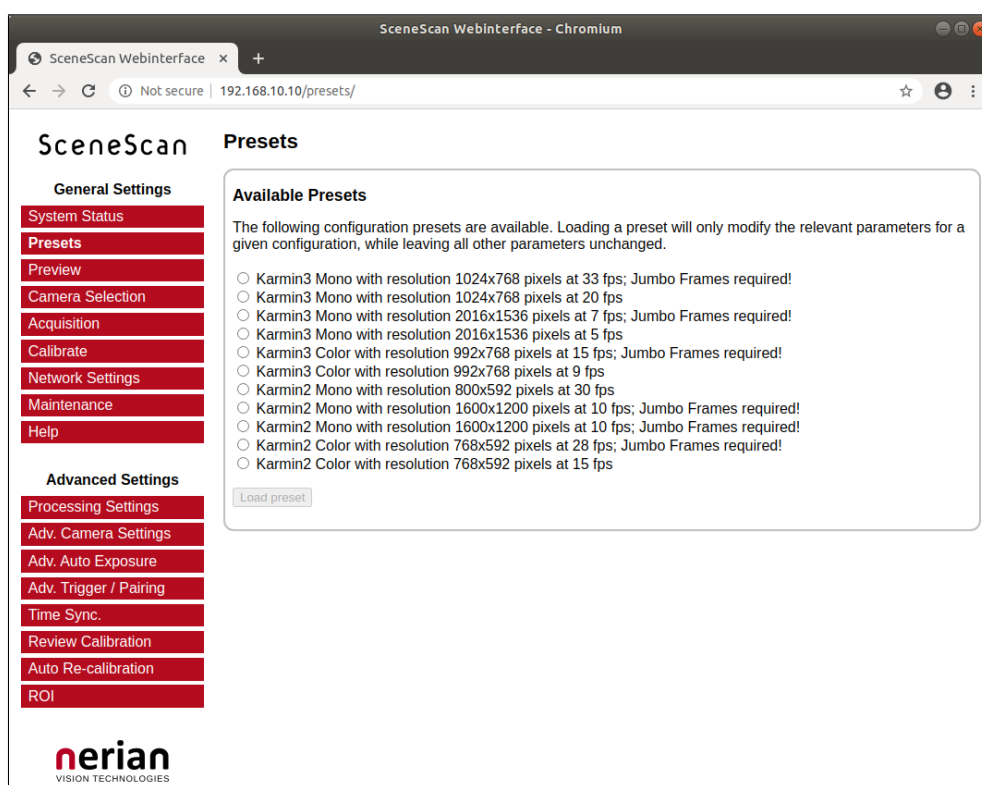
19

Figure 13: Screenshot of configuration presets page.

accessing the cameras, or another system error might have occurred. Please consult the system logs in this case. The image processing subsystem will be started immediately once the cause of error has been removed (such as connecting the cameras).

SOC temperature: The temperature of the central System-on-Chip (SoC) that performs all processing tasks. The maximum operating temperature for the employed SoC is at 100 °C. A green-orange-red color-coding is applied to signal good, alarming and critical temperatures.

System logs: List of system log messages sorted by time. In regular operation, you will find information on the current system performance. In case of errors, the system logs contain corresponding error messages.

9.2 Presets

Different configuration presets are available in case SceneScan is used with one of Nerian's Karmin2 or Karmin3 stereo cameras. The use of a preset is highly recommended when working with one of these stereo cameras. For third-party cameras, configuration presets are not available.

Figure 13 shows the *presets* web-interface page. Loading a preset will only modify the parameters that are relevant for a given configuration. Other pa-
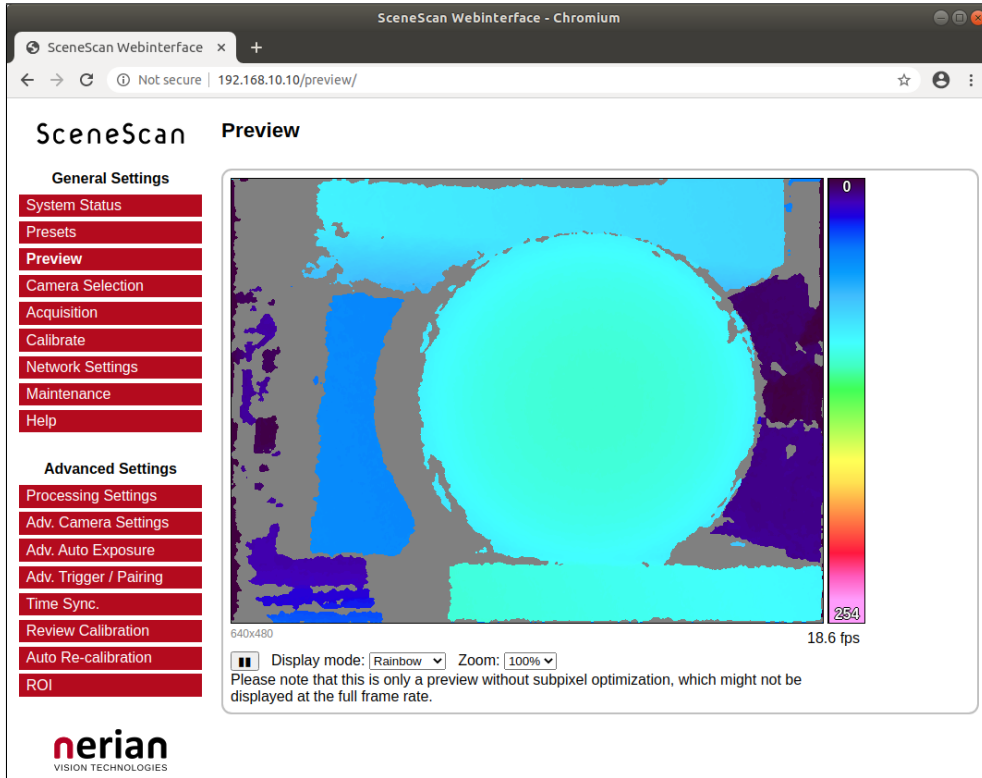
Figure 14: Screenshot of configuration preview page.

rameters will not be modified. If all parameters should be set to the preferred default value, it is recommended to first perform a configuration reset (see Section 9.8) and then load the desired preset afterwards.

9.3 Preview

The *preview* page, which is shown in Figure 14, provides a live preview of the currently computed disparity map. Please make sure that your network connection supports the high bandwidth that is required for streaming video data (see Section 7.1). For using the preview page, you require a direct network connection to SceneScan. An in-between proxy server or a router that performs network address translation (NAT) cannot be used.

When opening the preview page, SceneScan stops transferring image data to any other host. The transfer is continued as soon as the browser window is closed, the user presses the pause button below the preview area, or if the user navigates to a different page. Only one open instance of the preview page, or any other page that is streaming video data to the browser, is allowed at a time. If attempted to open more than once, only one instance will receive data.

The preview that is displayed in the browser does not reflect the full quality of the computed disparity map. In particular, the frame rate is limited to 20
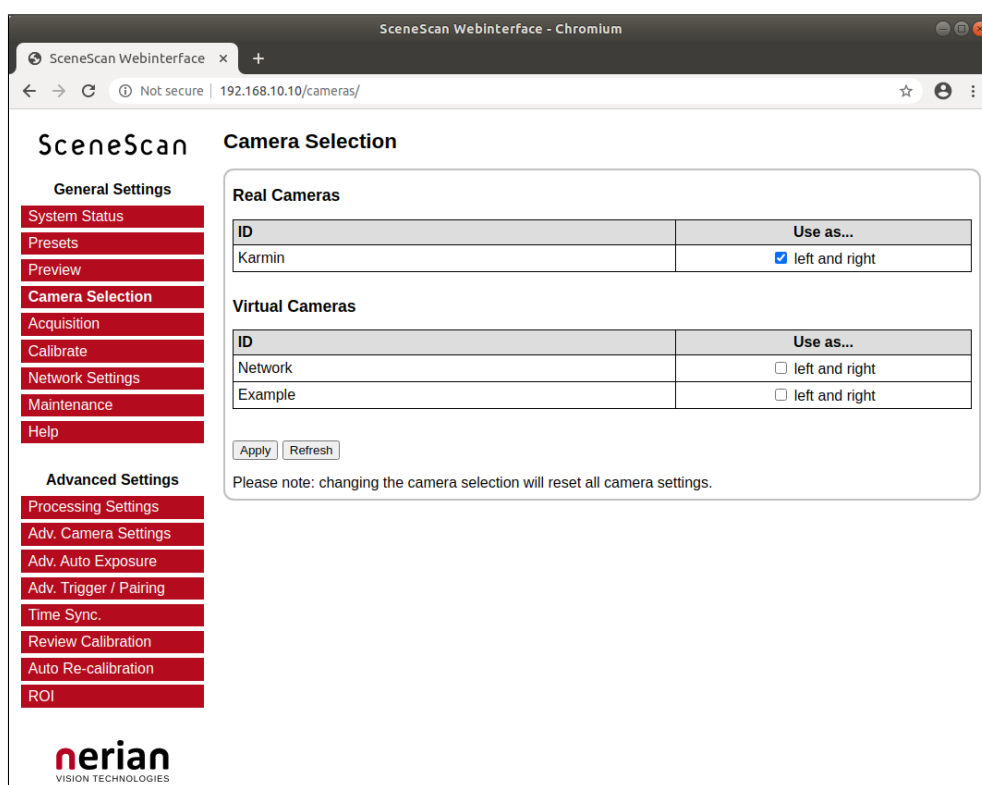
Figure 15: Screenshot of configuration page for camera settings.

fps and sub-pixel accuracy is not available. To receive a full-quality preview, please use the NVCom application, which is described in Section 11.1.

Different color-coding schemes can be selected through the drop-down list below the preview area. A color scale is shown to the right, which provides information on the mapping between colors and disparity values. The possible color schemes are:

Red / blue: A gradient from red to blue, with red hues corresponding to high disparities and blue hues corresponding to low disparities. Invalid disparities are depicted in black.

Rainbow: A rainbow color scheme with low wavelengths corresponding to high disparities and high wavelengths corresponding to low disparities. Invalid disparities are depicted in gray.

Raw data: The raw disparity data without color-coding. The pixel intensity matches the integer component of the measured disparity. Invalid disparities are displayed in white.

9.4   Camera Selection

The *camera selection* page that is shown in Figure 15 allows for the selection of a desired stereo camera or camera pair. All detected physical cameras are

listed in the *'Real Cameras'* list. In the *'Virtual Cameras'* list there are the two virtual stereo cameras that were mentioned in Section 7.6, which provide an example stereo sequence or facilitate the reception of input images through ethernet.

To select one of Nerian's Karmin2 or Karmin3 stereo cameras, one needs to tick the *'left and right'* check box. If two separate cameras shall be used instead, one needs to manually select which camera shall be the left and which one the right camera. This ordering must be done from the camera's perspective. If the left and right cameras are mixed up, this can be automatically detected during camera calibration and the cameras will be automatically reassigned.

Please be aware that changing the camera selection will reset all camera parameters. This also includes the cameras' calibration data. Hence a re-calibration will be necessary (see Section 9.6).

9.5   Acquisition Settings

The most relevant parameters for image acquisition are listed on the *acquisition settings* page that is shown in Figure 16. This page is divided into three distinct areas.

9.5.1   Simple Camera Settings

Settings that are provided by the attached cameras are listed in the *'simple camera settings'* area. Adjusting any of the given parameters will change the configuration for both cameras equally. Please note that the *apply* button must be pressed in order for any configuration changes to become effective. By pressing the *'reset camera defaults'* button, you can reset the cameras' settings to the default configuration. This is usually the configuration that has been written to the cameras' internal memory through the manufacturer software. If the *reset* button is pressed, all configuration changes that have not been confirmed through the *apply* button are reverted.

Exactly which parameters are displayed depends on the connected cameras. For a detailed explanation of these parameters, we thus recommend consulting the camera manufacturer's manual. With a typical stereo camera or individual camera pair, the parameters described below should be available.

Image Format Control

Binning horizontal: Number of horizontal photosensitive cells that are combined for one image pixel. There might be conditional dependencies to the *binning vertical* parameter, which might have to be adjusted first.

Binning vertical: Number of vertical photosensitive cells that are combined for one image pixel. There might be condi-
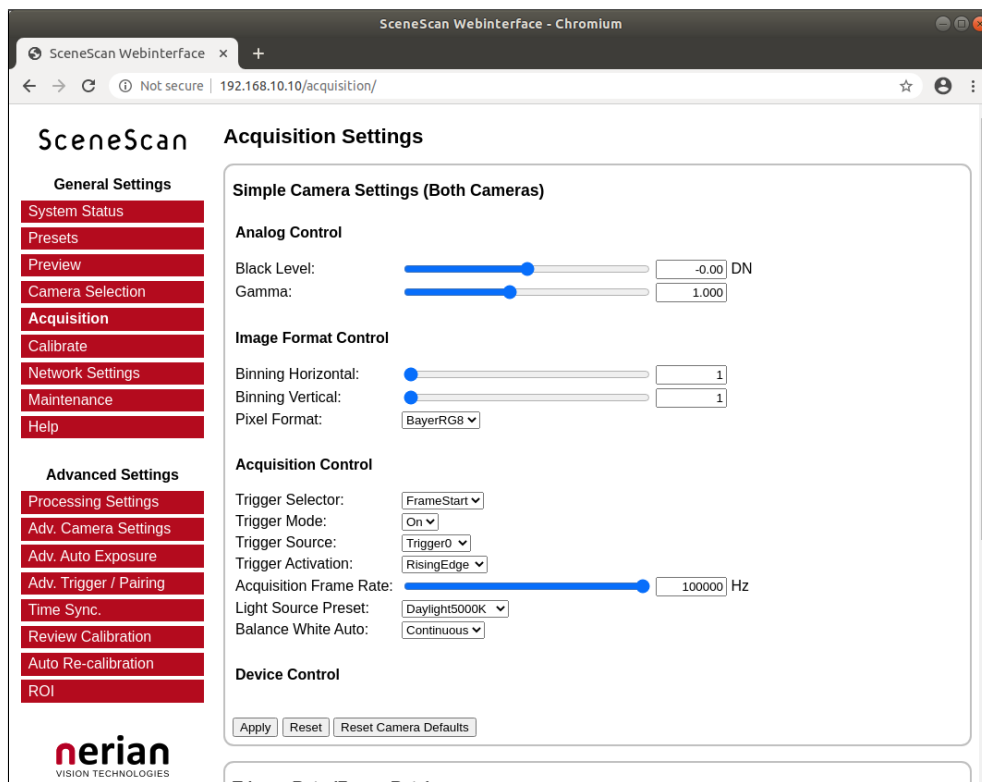
Figure 16: Screenshot of configuration page for acquisition settings.

	tional dependencies to the *binning horizontal* parameter, which might have to be adjusted first.
Pixel format:	Desired pixel encoding mode. For supported formats see Section 4.2.

Analog Control

Black level:	Controls the analog black level as an absolute physical value.
Gamma:	Controls the gamma correction of pixel intensity.

Acquisition Control

Trigger selector:	Selects the trigger that shall be configured.
Trigger mode:	Controls if the selected trigger is active.
Trigger source:	Specifies the internal signal or physical input line to use as the trigger source.
Trigger activation:	Specifies the activation mode of the trigger.

24

9.5.2   Trigger Rate / Frame Rate

Cameras that are connected to SceneScan should always be triggered externally (see Section 7.7). Hence the cameras' frame rate settings do not have an effect and should be left at maximum. Rather, the trigger rate should be adjusted in order to control the effective frame rate. This can be done in the *'trigger rate (frame rate)'* area. More detailed trigger options are available on the *'advanced trigger / pairing settings'* page (see Section 9.12), in case a detailed control of the trigger waveform is required.

9.5.3   Exposure Control

SceneScan will automatically control the cameras' exposure and gain to match a given average intensity, which can be selected in the *'exposure control'* area. If an automatic adjustment is not desired, then the user can alternatively specify a manual exposure time and gain setting. More advanced exposure and gain options are available on the *'advanced auto exposure and gain settings'* page (see Section 9.11).

9.6   Camera Calibration

9.6.1   Calibration Board

The *'calibrate cameras'* page, which is shown in Figure 17, enables the calibration of the stereo camera pair. You require a calibration board, which is a flat panel with a visible calibration pattern on one side. The pattern that is used by SceneScan consists of an asymmetric grid of black circles on a white background, as shown in Figure 18.

The pattern can be downloaded directly from the calibration page. Simply select the desired pattern size in the *'calibration board'* drop-down list, and click the download link. Even if you already have a calibration board, make sure to always select the correct size before starting the calibration process. Otherwise the calibration results cannot be used for 3D reconstruction with a correct metric scale (see Section 8.2).

Should you require a calibration board with a custom size, then you can select *custom* from the *'calibration board'* drop-down list. This allows you to enter the calibration board details manually. The first dimension of the *pattern size* is the number of circles in one grid column. This number must be equal for all columns of the circles grid.

The number of circles per row is allowed to vary by 1 between odd and even rows. The second dimension is thus the sum of circles in two consecutive rows. All downloadable default calibration patterns have a size of 4×11.

The last parameter that you have to enter when using a custom calibration board is the *circle spacing.* This is the distance between the centers of two neighboring circles. The distance must be equal in horizontal and vertical direction for all circles.
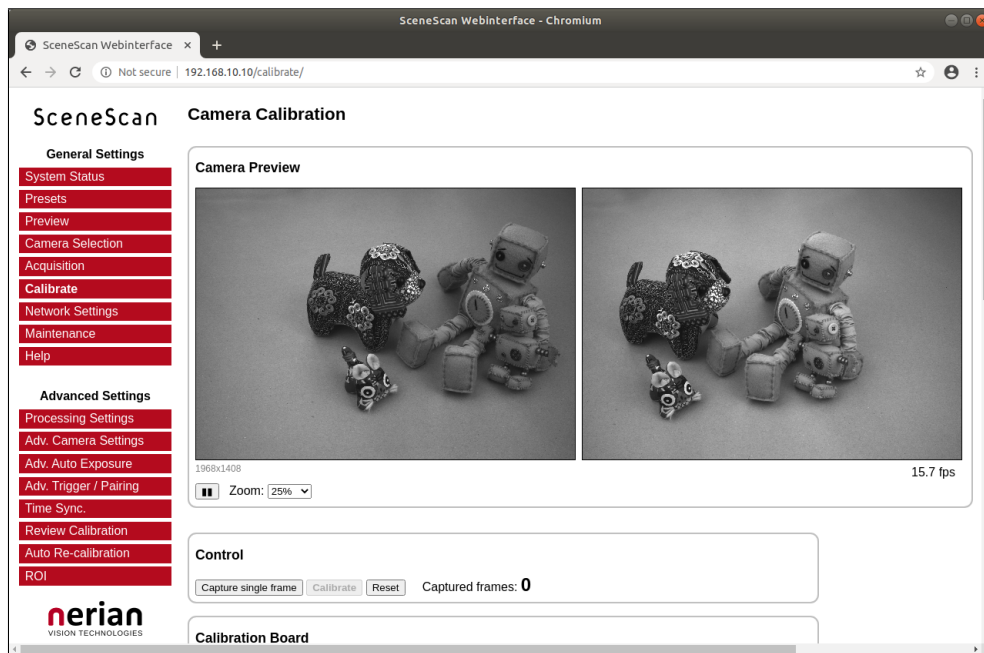
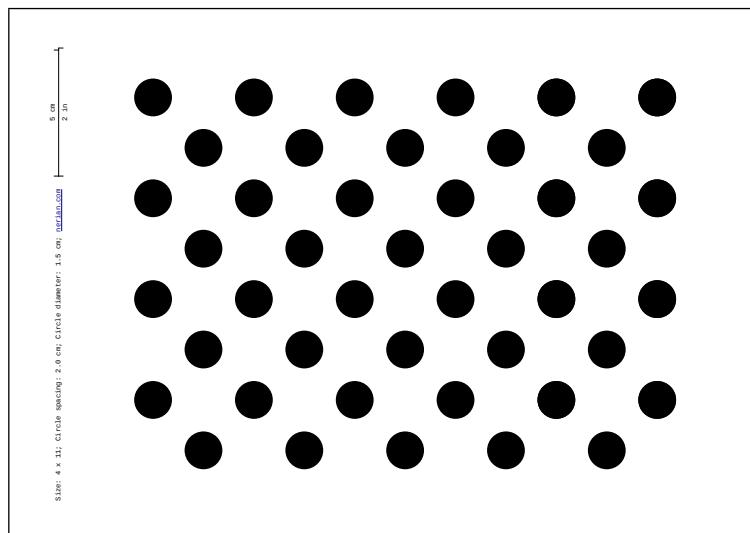Figure 17: Screenshot of configuration page for camera calibration.

Figure 18: Calibration board used by SceneScan.

9.6.2   Recording Calibration Frames

Once the calibration board settings have been set, you can start recording calibration frames. A live preview of both cameras is displayed in the *'camera preview'* area. The camera resolution during calibration is set to the maximum valid image size for the currently active image format and acquisition settings. Make sure that the calibration board is fully visible in both camera images and then press the *'capture single frame'* button in the *control* section. Repeat this process several times while moving either the camera or the calibration board.

The calibration board must be recorded at several different positions and orientations. You should also vary the distance of the board to the cameras and make sure that you cover most of the cameras' field of view. When recording the calibration frames, it is important that both cameras are synchronized.

The more frames you record, the more accurate the computed calibration will be. However, more frames also cause the computation of the calibration parameters to take longer. SceneScan supports the recording of up to 40 calibration frames. We recommend using at least 20 calibration frames in order to receive accurate results.

The recording of calibration frames can be simplified by activating the *'auto capture'* mode. In this mode, a new calibration frame is recorded in fix *capture intervals*. You can enter the desired interval in the auto capture section and then press the *'start auto capture'* button. If desired, an audible sound can be played to signal the countdown and the recording of a new frame. Auto capture mode can be stopped by pressing the *'stop auto capture'* button.

A small preview of each captured calibration frame is added to the *'captured frames'* section. The frames are overlaid with the detected positions of the calibration board circles. You can click any of the preview images to see the calibration frame at its full resolution. An example for a calibration frame with a correctly detected calibration board is shown in Figure 19. If the calibration board was not detected correctly or if you are unhappy with the quality of a calibration frame, then you can delete it by clicking on the ×-symbol.

9.6.3   Performing Calibration

Once you have recorded a sufficient number of calibration frames, you can initiate the calibration process by pressing the *calibrate* button in the *control* section. The time required for camera calibration depends on the number of calibration frames that you have recorded. Calibration will usually take several minutes to complete. If calibration is successful then you are immediately redirected to the *'review calibration'* page.

Calibration will fail if the computed vertical or horizontal pixel displacement exceeds the allowed range of $[-39, +39]$ pixels for any image point. The most common causes for calibration failures are:

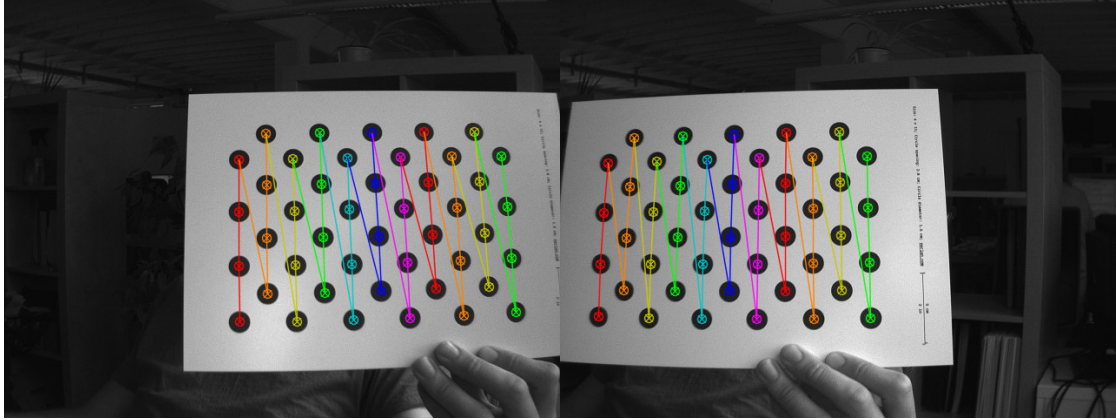- Insufficient number of calibration frames.

Figure 19: Example calibration frame with detected calibration board.

- Improperly aligned cameras. See Section 7.5.

- Lenses with strong geometric distortions.

- Lenses with unequal focal lengths.

- Improper camera synchronization.

- Frames with calibration board misdetections.

Should calibration fail, then please resolve the cause of error and repeat the calibration process. If the cause of error is one or more erroneous calibration frames, then you can delete those frames and re-press the *calibrate* button. Likewise, in case of too few calibration frames, you can record additional frames and restart the calibration computation.

9.6.4   Reviewing the Region of Interest

The calibration process can determine that an offset for the cameras' regions of interest (ROIs) has to be imposed to correct for relative sensor displacement. This also constrains the available ROI size. In case a ROI offset has been set, a message will pop up after successful calibration, from which you can proceed to the ROI selection page (see Section 9.16) to revise the region of interest before proceeding.

9.7   Network Settings

The *'network settings'* page, which is displayed in Figure 20, is used for configuring all network related parameters.

SceneScan can query network configuration automatically via DHCP client requests, which are enabled by default to aid switching between existing network setups. SceneScan devices in a network that assigns IP settings through
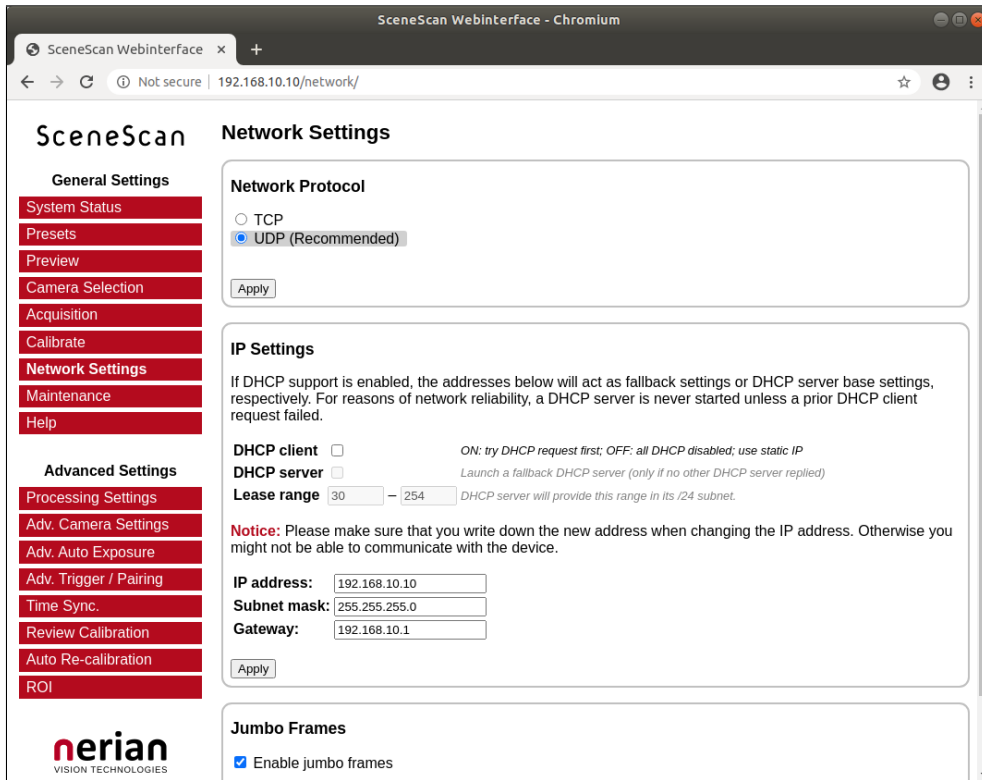
Figure 20: Screenshot of configuration page for network settings.

DHCP are easily discovered and accessed via the device discovery API and also the NVCom utility (Section 11.1). If no DHCP servers are present, Scene-Scan uses its static IP settings as a fallback. DHCP client support can be disabled if fixed IP settings are desired and the device will not be switched between different networks. This will speed up device initialization (reboot) time by approximately 3 seconds, the total DHCP wait time. In this case, the IP settings in this section are used as static values.

SceneScan also contains a fallback DHCP server. It is enabled by default but only launched when a prior DHCP client request failed. This means that no DHCP server is ever launched if DHCP client support is turned off, to ensure that SceneScan will never compete with an existing DHCP server. The SceneScan DHCP server uses the IP address settings as a base; the lease range is always in the /24 subnet of the IP address.

In the *'IP settings'* section, you can disable or enable the DHCP components and specify an IP address, subnet mask and gateway address, which are used as static configuration or fallback configuration depending on the DHCP settings. When changing the IP settings, please make sure that your computer is in the same subnet, or that there exists a gateway router through which data can be transferred between both subnets. Otherwise you will not be able to access the web interface anymore and you might be forced to perform a configuration reset (see Section 6.4).
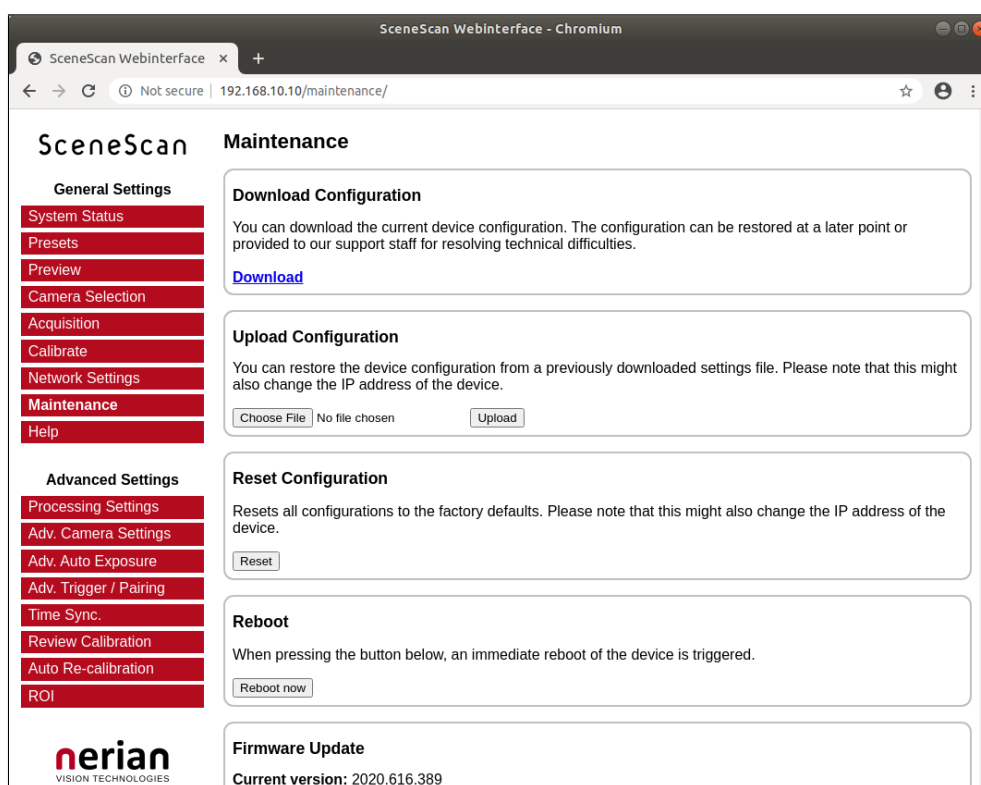
Figure 21: Screenshot of configuration maintenance page.

In the *'network protocol'* section, you can choose the underlying network protocol that shall be used for delivering the computation results to the client computer. The possible options are *TCP* and *UDP*. Due to the high-bandwidth real time data we recommend using UDP, unless the input images are transferred through ethernet, as described in Section 7.6.

In order to obtain the best possible performance, jumbo frames support should be activated in the *'jumbo frames'* section. Before doing so, however, you must make sure that jumbo frames support is also enabled for your client computer's network interface. Details on how to enable jumbo frame support on your computer can be found in Section 7.2 on page 10. For Linux client computers, the jumbo frames (MTU) setting is automatically applied when receiving configuration from an active SceneScan DHCP server. Please note that in this case changing the SceneScan Jumbo Frames mode or MTU Size necessitates new DHCP leases to propagate the setting (e.g. by unplugging and re-inserting the network cable).

9.8   Maintenance

On the *maintenance* page that is shown in Figure 21, you can download a file that contains the current device configuration and the system logs, by pressing the *download* link. In case of technical problems please include this file in your

support request, such that your device configuration can be reproduced and that system problems can be investigated.

A downloaded configuration file can be re-uploaded at a later point in time. This allows for a quick switching between different device configurations. In order to upload a configuration, please select the configuration file and press the *upload* button. Please be aware that uploading a different configuration might modify the IP address of the device. In order to avoid a faulty configuration state, please only upload configurations that have previously been downloaded through the web interface.

If you are experiencing troubles with your current device configuration, you can reset all configuration settings to the factory defaults, by pressing the *reset* button. Please note that this will also reset the network configuration, which might lead to a change of SceneScan's IP address. This is equivalent to pushing the reset button on the backside of the device (see Section 6.4).

If SceneScan shows signs of erroneous behavior, it is possible to reboot the device by pressing the *'reboot now'* button. It will take several seconds until a reboot is completed and SceneScan is providing measurement data again. Please use this function as an alternative to a power cycle, if the device cannot be easily accessed.

The maintenance page further allows you to perform firmware updates. Use this functionality only for firmware files that have officially been released by Nerian Vision Technologies. To perform a firmware update, select the desired firmware file and press the *update* button. The update process will take several seconds. **Do not unplug the device, reload the maintenance page or re-click the update button while performing firmware updates**. Otherwise, this might lead to a corrupted firmware state. Once the update has been completed, please restart the device by briefly unplugging the power supply.

9.9 Processing Settings

9.9.1 Operation Mode

The major processing parameters can be changed on the *'processing settings'* page, which is shown in Figure 22. The most relevant option is the operation mode, which can be set to one of the following values:

Pass through: In this mode SceneScan forwards the imagery of the selected cameras without modification. This mode is intended for verifying that both cameras are functioning correctly.

Rectify: In this mode SceneScan transmits the rectified images of both cameras. This mode is intended for verifying the correctness of the image rectification.
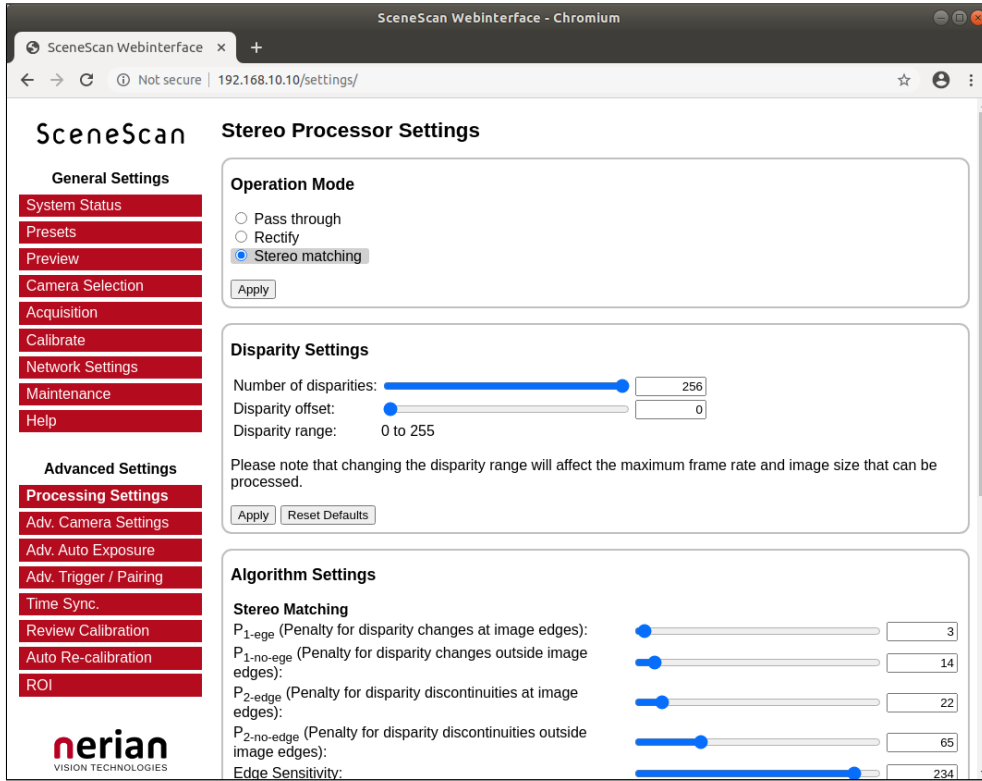
Figure 22: Screenshot of configuration page for processing settings.

Stereo matching: This is the default mode, in which SceneScan performs the actual stereo image processing (stereo matching). SceneScan transmits the rectified image of the left camera and the left camera disparity map.

9.9.2 Disparity Settings

If the operation mode is set to stereo matching, then the *'disparity settings'* allow for a configuration of the disparity range that is searched by SceneScan. The disparity range affects the frame rate that can be achieved. **The frame rate should be adjusted once the disparity range has been changed** (see Section 4.4 on page 5 for recommendations). Please be aware that increasing the disparity range will also reduce the maximum image size that can be configured.

The *'number of disparities'* option specifies the total number of pixels that are searched for correspondences. This option has a high impact on the depth resolution and the covered measurement range (see Section 8.2). The start of the disparity range can be chosen through the *'disparity offset'* option. Typically, a value of 0 is desired for the offset, which allows for range measurements up to infinity. If the observable distance is certain to be constrained, then low disparity values won't occur. In this case it is possible to increase the disparity offset, such that these low disparities are not computed.

9.9.3 Algorithm Settings

The behavior of the image processing algorithms can be controlled through the *algorithm settings*. The default configuration has been determined using machine learning methods, and it should thus be the best choice for most use cases. Nevertheless, all algorithm parameters can be adjusted through the web interface. The following parameters control the stereo matching algorithm:

Penalty for disparity changes (P_1): A penalty that is applied to gradually changing disparities. A large value causes gradual disparity changes to occur less frequently, while a small value causes gradual changes to occur more frequently. Different values can be configured for pixels that are on image edges (P_{1-edge}) and pixels that are not on edges ($P_{1-no-edge}$). These values must be smaller than the values for P_2.

Penalty for disparity discontinuities (P_2): A penalty that is applied to abruptly changing disparities. A large value causes disparity discontinuities to occur less frequently, while a small value causes discontinuities to occur more frequently. Different values can be configured for pixels that are on image edges (P_{2-edge}) and pixels that are not on edges ($P_{2-no-edge}$). These values must be greater than the values for P_1.

SceneScan applies an optimization algorithm to improve the accuracy of the computed disparity map to sub-pixel resolution. If only a small region of interest (ROI) of the input image / disparity map is relevant, then this auto-tuning process can be constrained to only this ROI. In this case one should expect more accurate sub-pixel measurements inside the ROI. The relevant parameters for constraining the sub-pixel tuning ROI are:

Tune sub-pixel optimization on ROI: If enabled, the sub-pixel optimization is tuned on the region defined by the subsequent parameters, instead of the whole image.

Width: Width in pixels of the selected Region of Interest (ROI).

Height: Height in pixels of the selected ROI.

Offset X: Horizontal offset of the ROI relative to the image center.

Offset Y: Vertical offset of the ROI relative to the image center.

SceneScan implements several methods for post-processing the computed disparity map. Each post-processing method can be activated or deactivated individually. The available methods are:

Mask border pixels: If enabled, this option marks all disparities that are close to the border of the visible image area as invalid, as they have a high uncertainty. This also includes all pixels for which no actual image data is available, due to the warping applied by the image rectification (see Section 8.1).

Consistency check: If enabled, stereo matching is performed in both matching directions, left-to-right and right-to-left. Pixels for which the disparity is not consistent are marked as invalid. The sensitivity of the consistency check can be controlled through the *'consistency check sensitivity'* slider.

Uniqueness check: If enabled, pixels in the disparity map are marked as invalid if there is no sufficiently unique solution (i.e. the cost function does not have a global minimum that is significantly lower than all other local minima). The sensitivity of the uniqueness check can be controlled through the *'uniqueness check sensitivity'* slider.

Texture filter: If enabled, pixels that belong to image regions with little texture are marked as invalid in the disparity map, as there is a high likelihood that these pixels are mismatched. The sensitivity of this filter can be adjusted through the *'texture filter sensitivity'* slider.

Gap interpolation: If enabled, small patches of invalid disparities, which are caused by one of the preceding filters, are filled through interpolation.

Noise reduction: If enabled, an image filter is applied to the disparity map, which reduces noise and removes outliers.

Speckle filter iterations: Marks small isolated patches of similar disparity as invalid. Such *speckles* are often the result of erroneous matches. The number of iterations specify how aggressive the filter will be with removing speckles. A value of 0 disables the filter.

9.10   Advanced Camera Settings

More advanced camera parameters can be adjusted on the *'advanced camera settings'* page, which is shown in Figure 23. This page provides two separate configuration areas, one for the left camera and one for the right camera, which can be adjusted individually.

Most machine vision cameras provide a large number of parameters that can be adjusted. In order to keep the configuration manageable, the parameters are sorted into different visibility groups. The visibility group can be changed through the drop-down list in the top right corner of each configuration area. By default the *Beginner* visibility group is selected, which only contains the most basic features. In order to view more advanced settings, please select the *Expert* or *Guru* visibility group.

For an explanation of the various parameters, please refer to the documentation from your camera manufacturer.
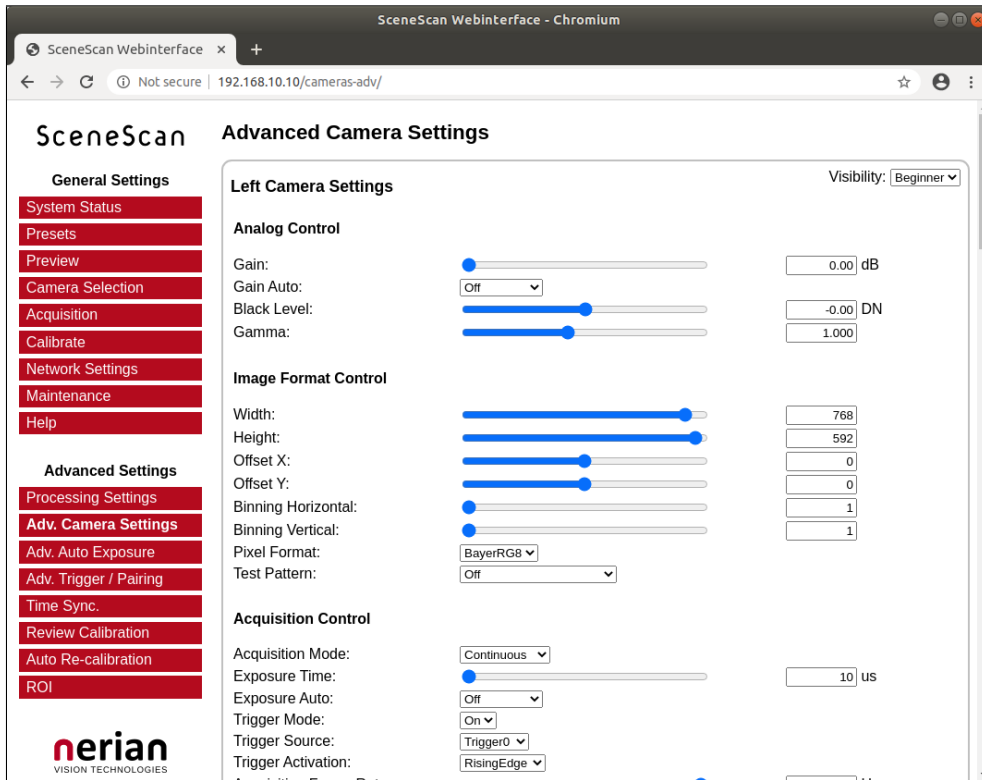
Figure 23: Screenshot of configuration page for advanced camera settings.

9.11   Advanced Automatic Exposure and Gain Settings

To ensure the best possible image quality, SceneScan provides a fully automatic exposure time and gain adaptation for rapidly changing lighting conditions, which often occurs in outdoor scenarios. You can activate and deactivate both auto functions independently on the *auto exposure* page, which is shown in Figure 24.

9.11.1   Exposure and Gain

Mode: Selects whether exposure time and/or gain are adjusted automatically. Under normal circumstances *'auto exposure and gain'* should be selected for the automatic adjustment of both parameters.

Target intensity: Selects an average intensity value for the stereo images, which is targeted by the automatic adjustment. Intensity values are written in percentage numbers with 0 representing black and 100 white.

Intensity delta: If the average intensity of the image is in the range of *intensity delta* to the *target intensity*, no adjustment is performed, no matter what mode is selected. You can increase the value if you observe flickering in your image stream and decrease it if you need smoother variations between consecutive frames.
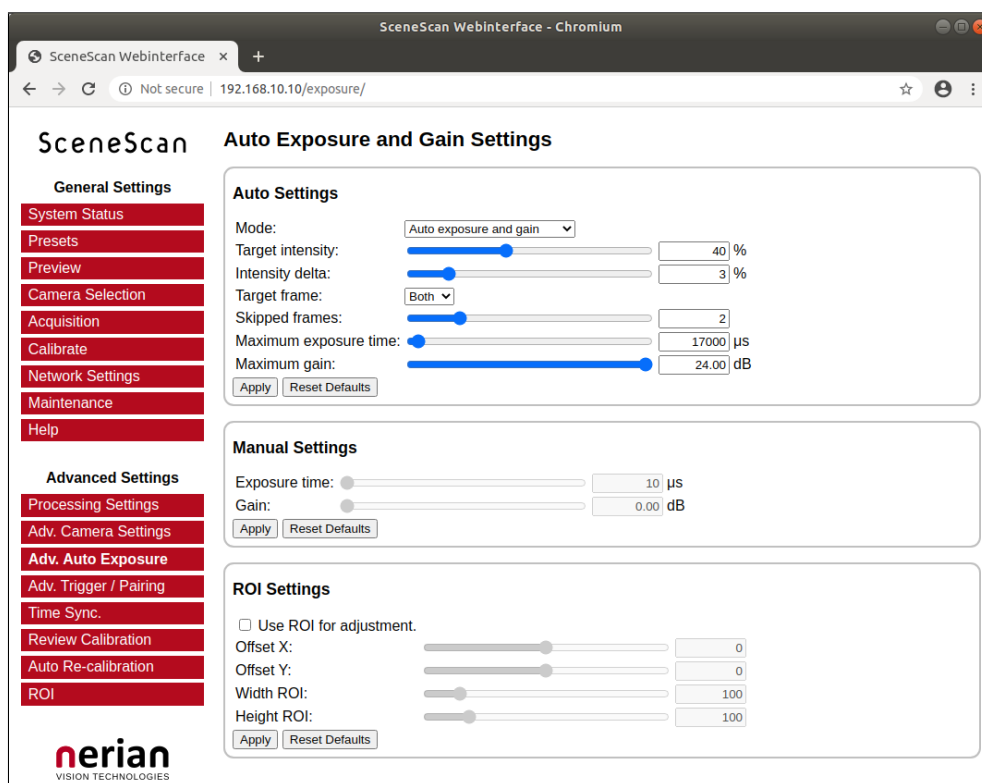
Figure 24: Screenshot of the configuration page for the automatic exposure and gain adjustment settings.

Target frame: Selects if the intensity of the *left* frame, the intensity of the *right* frame or the average intensity of *both* frames should be adjusted to the *target intensity*.

Skipped frames: The number of ignored frames between two adjustment steps. Increase if you want to reach a higher absolute frame rate. As most cameras have a delay between receiving a new parameter and applying it, that value should in general be a positive number and not set to zero.

Maximum exposure time: Depending on your selected frame rate, you might need to decrease the maximum exposure time to reach the desired rate. The value for the *maximum exposure time* should always be smaller than the time between two frames.

Maximum gain: Just like for the exposure time, it is also possible to constrain the maximum allowed gain. Constraining the gain can improve image processing results for situations with high sensor noise.

9.11.2 Manual Settings

If the automatic adjustment is deactivated in the *mode* selection, the exposure time and/or gain can be manually set to fixed values in this section.
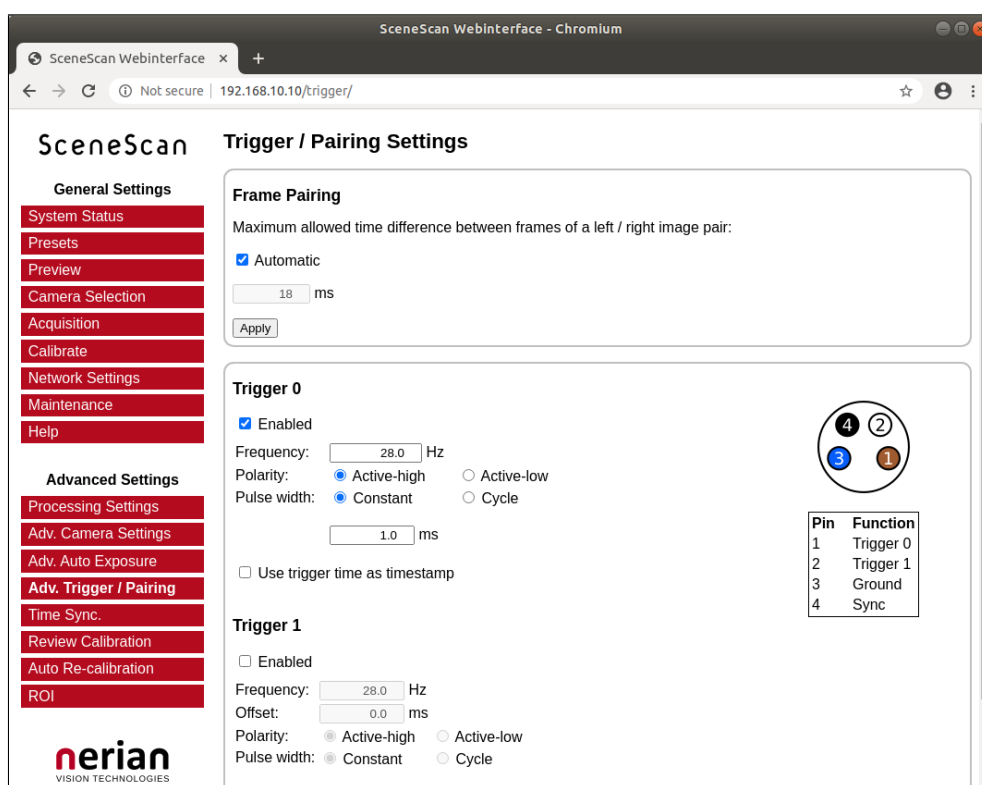
36

Figure 25: Screenshot of configuration page for trigger settings.

9.11.3 ROI Settings

Rather than performing the adjustment with respect to the average intensity of the complete image, you can compute the average intensity only on a region of interest. Enable *'use ROI for adjustment'* in that case. *'Offset X'* and *'Offset Y'* describe the region's center position relative to the image center. *'Width ROI'* and *'Height ROI'* let you adjust the spatial extension of the ROI. The ROI must be completely contained in the image. If this is not the case, the ROI will be cropped automatically.

9.12 Trigger / Pairing

The *'trigger / pairing'* page that is shown in Figure 25 allows for a configuration of the trigger output and frame pairing settings. Frame pairing refers to the process of identifying which left and right camera frames were recorded at the same time. This is done by comparing the timestamps at which the frames are received by SceneScan against a *maximum time difference.* Only frames whose time stamps do not differ by more than this maximum difference can form an image pair. By default this time difference is automatically determined, but if the automatic checkbox is unchecked a manual value can be entered.

If the cameras provide high resolution images or if they have a low pixel transfer speed, it is recommended to increase the maximum time difference.

As a general guideline, we recommend setting half of the time delay between two frames as the maximum time difference.

This page also allows for a configuration of SceneScan's trigger port. As described in Section 7.7, SceneScan features a trigger port that provides access to up to two trigger signals. The two trigger signals, *Trigger 0* and *Trigger 1*, can be enabled or disabled by selecting the respective check boxes.

For *Trigger 0* it is possible to select a frequency between 0.1 and 200 Hz and an arbitrary pulse width in milliseconds. The polarity of the generated trigger signal can be either active-high or active-low. The pulse width can be constant or cycle between a list of pre-configured values. A cycling pulse width configuration is usually used for high-dynamic-range (HDR) imaging. Some cameras such as Karmin3 have the ability to control the image exposure time through the trigger pulse width, exposing the sesnor for as long as the trigger signal is high. If the pulse width cycles between different values, then the cameras' exposure times will also cycle.

If the checkbox *'use trigger time as timestamp'* is selected, then the trigger time is transmitted as timestamp with each processing result. Please make sure that the cameras do not skip any trigger signals, as in this case the timestamp correlation will fail. This functionality should not be used with virtual cameras such as the hard-coded example and network capturing.

The signal *Trigger 1* can only be enabled if *Trigger 0* is also enabled. The frequency is forced to the same value as for *Trigger 0*. However, it is possible to specify a time offset, which is the delay from a leading edge of *Trigger 0* to a leading edge of *Trigger 1*. Furthermore, *Trigger 1* can have a pulse width and polarity configuration that differs from *Trigger 0*.

9.13   Time Synchronization

The *'time synchronization'* page, which is shown in Figure 26, can be used to configure three possible methods for synchronizing SceneScan's internal clock. As explained in Section 8.3, the internal clock is used for timestamping captured camera frames.

The first option is to synchronize with a time server, using the Network Time Protocol (NTP) up to version 4. In this case SceneScan synchronizes its internal clock to the given time server, using Coordinated Universal Time (UTC). The accuracy of the time synchronization depends on the latency of your network and time server. If NTP time synchronization is active, synchronization statistics are displayed in a dedicated status area.

As an alternative to NTP, the Precision Time Protocol (PTP) can be used for synchronization. PTP provides a significantly higher accuracy when compared to NTP, and should hence be preferred if available. Like for NTP, the clock will also be set to UTC and synchronization status information will be displayed.

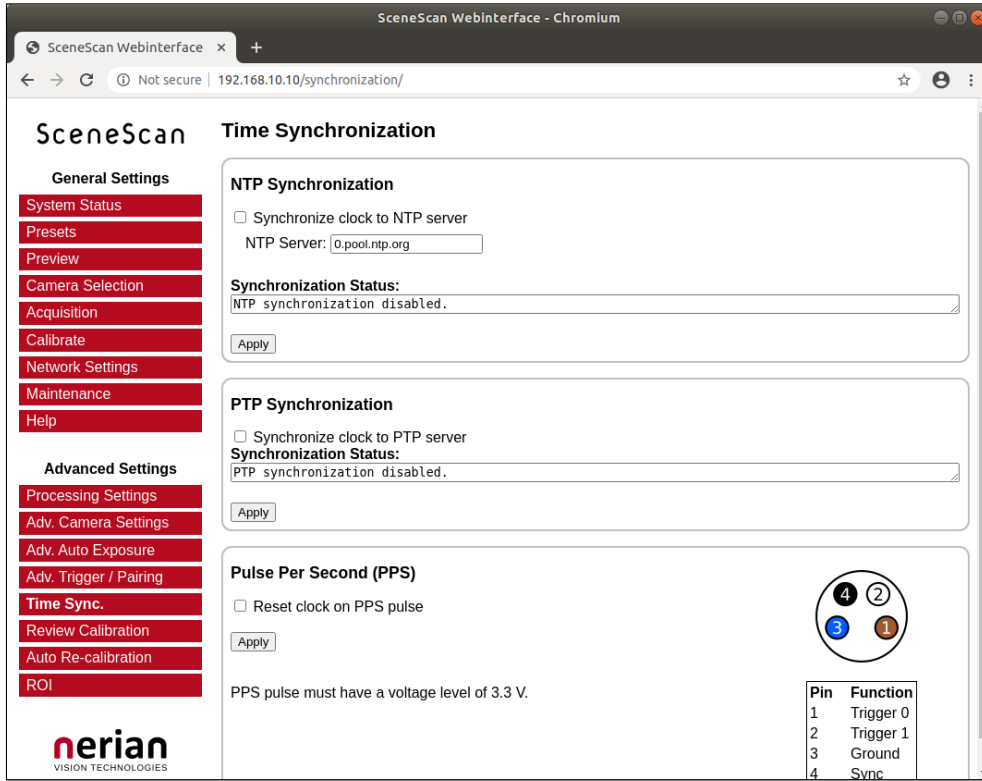When activating Pulse Per Second (PPS) synchronization, the internal

Figure 26: Screenshot of configuration page for time synchronization.

clock is set to 0 whenever a synchronization signal is received. Please refer to Section 7.8 on page 15 for details on the PPS synchronization.

9.14   Reviewing Calibration Results

Once calibration has been performed, you can inspect the calibration results on the *'review calibration'* page, which is shown in Figure 27. On the top of this page you can see a live preview of both cameras as they are rectified with the current calibration parameters. Please make sure that corresponding points in the images of both cameras have an identical vertical coordinate.

By activating the *'display epipolar lines'* option, you can overlay a set of horizontal lines on both camera images. This allows for an easy evaluation of whether the equal vertical coordinates criterion is met. An example for a left and right input image with overlaid epipolar lines is shown in Figure 28.

In the *'quality information'* section you can find the *average reprojection error*. This is a measure for the quality of your camera calibration, with lower values indicating better calibration results. Please make sure that the average reprojection error is well below 1 pixel.

All computed calibration parameters are displayed in the *'calibration data'* section. These parameters are:

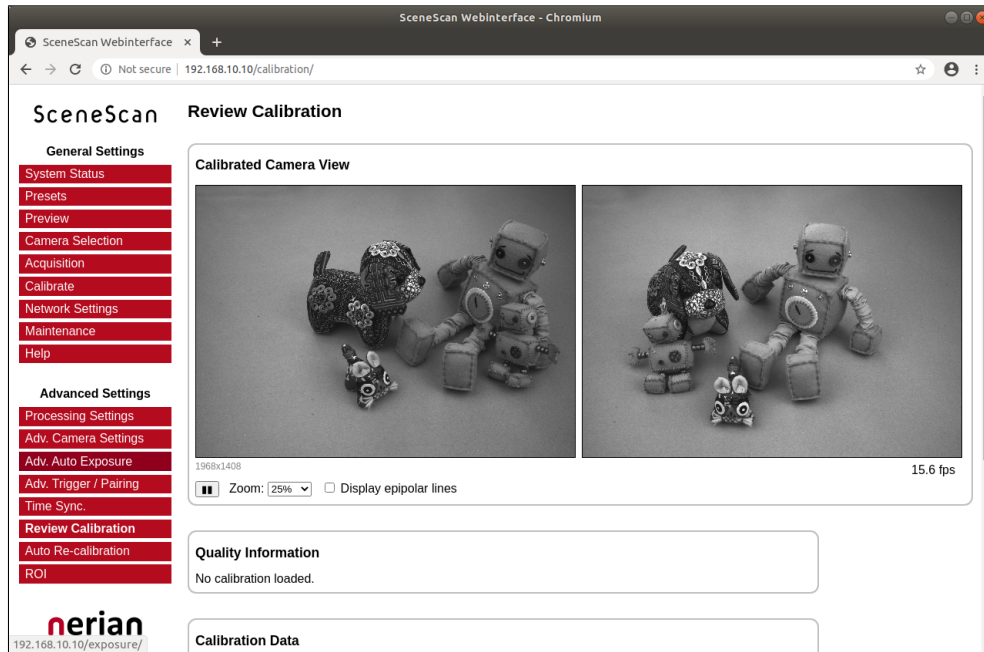**M₁ and M₂:** camera matrices for the left and right camera.

Figure 27: Screenshot of configuration page for reviewing camera calibration.
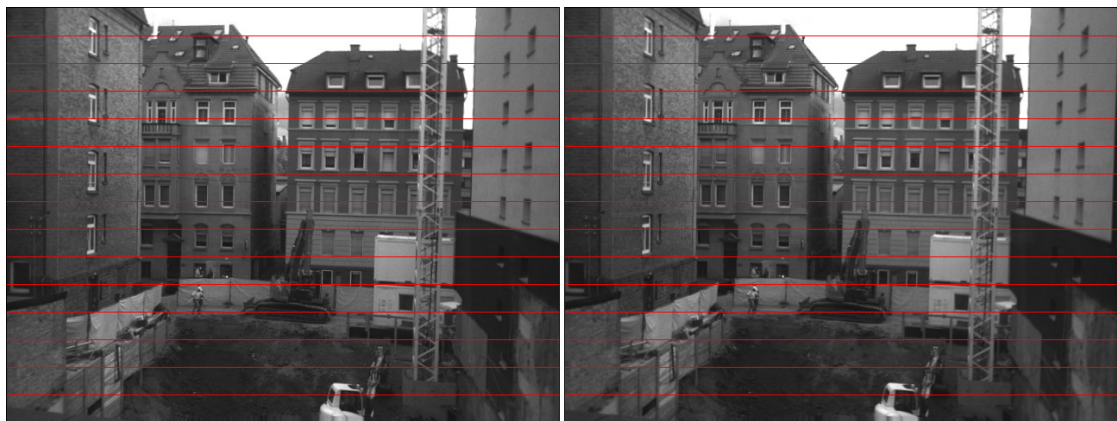

Figure 28: Example for evaluating vertical image coordinates.

D₁ and D₂: distortion coefficients for the left and right camera.

R₁ and R₂: rotation matrices for the rotations between the original and rectified camera images.

P₁ and P₂: projection matrices in the new (rectified) coordinate systems.

Q: the disparity-to-depth mapping matrix. See Section 8.2 for its use.

T: translation vector between the coordinate systems of both cameras.

R: rotation matrix between the coordinate systems of the left and right camera.

The camera matrices M_1 and M_2 are structured as follows:

$$M_i = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} , \tag{1}$$

where f_x and f_y are the lenses' focal lengths in horizontal and vertical direction (measured in pixels), and c_x and c_y are the image coordinates of the projection center.

The distortion coefficient vectors D_1 and D_2 have the following structure:

$$D_i = \begin{bmatrix} k_1 & k_2 & p_1 & p_2 & k_3 \end{bmatrix} , \tag{2}$$

where k_1, k_2 and k_3 are radial distortion coefficients, and p_1 and p_2 are tangential distortion coefficients.

You can download all calibration information as a machine-readable YAML file, by clicking the download link at the bottom of the *'calibration data'* section. This allows you to easily import the calibration data into your own applications. Furthermore, you can save the calibration data to your PC and reload it at a later time, by using the *'upload calibration data'* section. This allows you to switch between different cameras or optics without repeating the calibration process.

You can also perform a reset of the calibration data by pressing the *'reset calibration'* button. In this case, image rectification is disabled and the unmodified image data is passed on to the stereo matching algorithm. Use this option when selecting the already rectified virtual example camera, as explained in Section 7.6.

9.15   Auto Re-calibration

On the *'auto re-calibration'* page, which is shown in Figure 29, you can enable an automated estimation of the camera calibration parameters. In this case, the cameras remain calibrated even if their alignment is subject to variations.
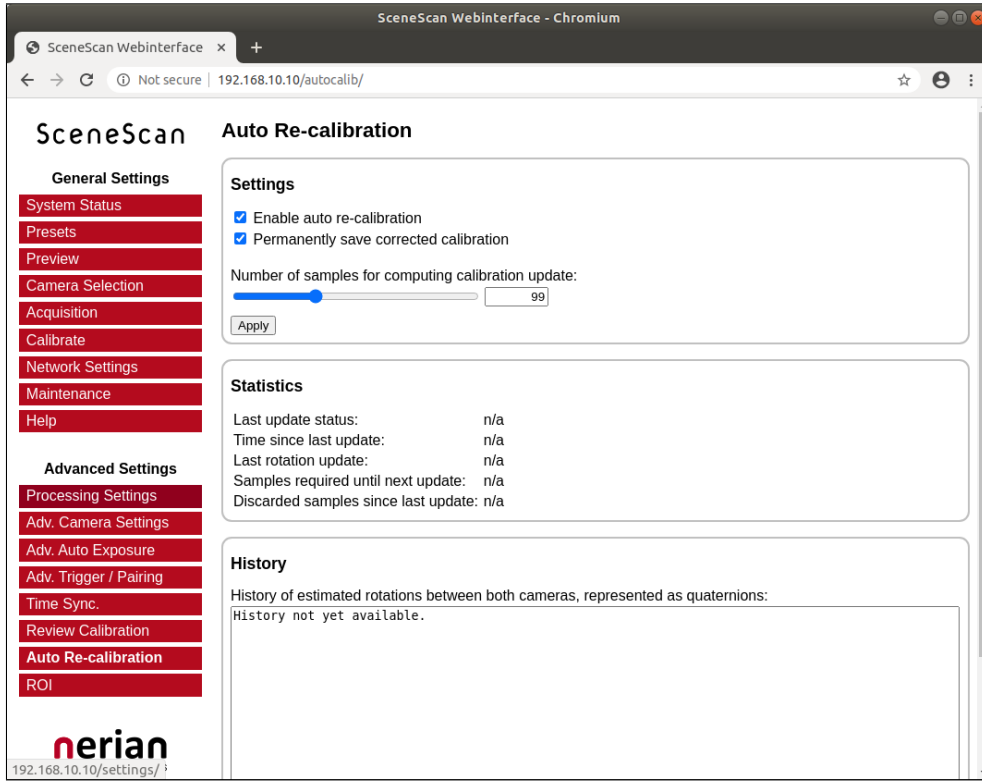
Figure 29: Screenshot of auto re-calibration settings.

For this process to work, it is necessary that the cameras have been calibrated once before with the manual calibration procedure (see Section 9.6).

Calibration parameters are usually divided into *intrinsic* parameters (focal length, projection center and distortion coefficients) and *extrinsic* parameters (transformation between the poses of both cameras). Auto re-calibration only performs an update of the extrinsic parameters, as they are significantly more prone to variations. More specifically, only the rotation between the cameras is estimated. This is usually the most fragile parameter, which can be affected significantly by even minor deformations of the camera mount.

Auto re-calibration can be activated by selecting the *'enable auto re-calibration'* option. SceneScan will then continuously compute samples for the estimated inter-camera rotation. A robust estimation method is applied for selecting a final rotation estimate from the set of rotation samples. The *number of samples* that are used for this estimation process can be configured. Small sample sizes allow for a quick reaction on alignment variations, while large sample sizes allow for very accurate estimates. If the *'permanently save corrected calibration'* option is selected, then the updated calibration is written to non-volatile memory and remains present even after a power cycle.

In the *statistics* area you can find various information on the current performance of the auto calibration process. This includes the status of the latest re-calibration attempt, the time since the last calibration update, the rota-
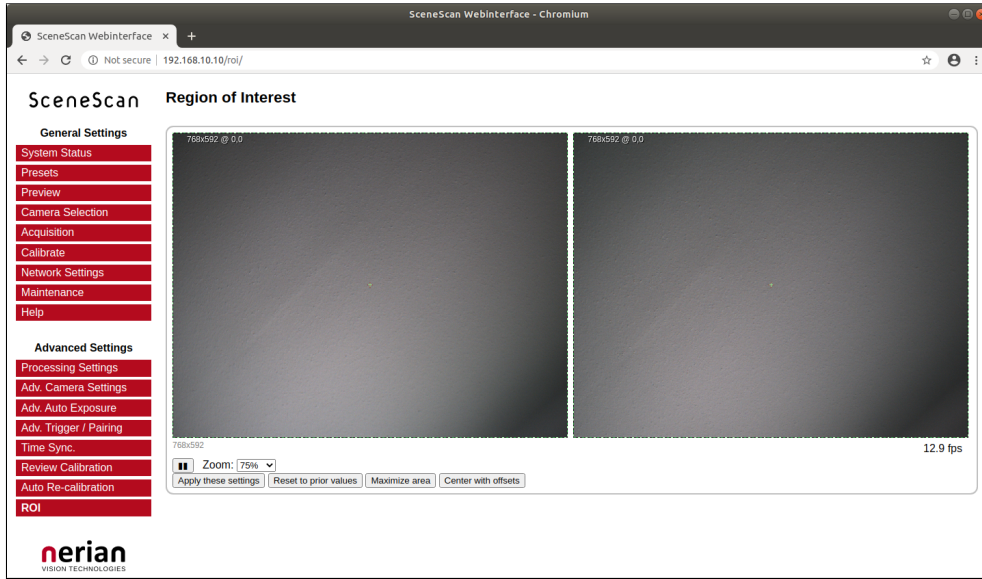
Figure 30: Screenshot of Region-of-Interest selection.

tional offset of the last update and the number of rotation samples that have been collected and discarded since the last update. Finally, you can find a list of recently computed inter-camera rotations in the *history* area. The listed rotations are represented as rotation quaternions.

9.16 Region of Interest

If not the entire camera image is needed but only a subsection, then this can be configured on the *region of interest* (ROI) page. This page will open the two camera images with overlaid frames showing the cropped region, which can be moved and resized in unison using the mouse (see Fig. 30). The device can revise the requested ROI dimensions; in this case you will see the region automatically snap to the closest valid image size.

It is advised to set the region of interest after first selecting the desired pixel format (see below) and then completing calibration, since the calibration procedure can impose a left-to-right ROI offset to correct for any displacements that are beyond the ranges handled by the hardware-accelerated rectification process.

10 API Usage Information

10.1 General Information

The cross-platform *libvisiontransfer* C++ and Python API is available for interfacing custom software with SceneScan. For Windows, a binary version of the library is available that can be used with Microsoft Visual Studio. For

43

Linux, please compile the library from the available source code. The API is included as part of the available software release, which can be downloaded from our support website[3].

The libvisiontransfer API provides functionally for receiving the processing results of SceneScan over a computer network. Furthermore, the API also allows for the transmission of image pairs. It can thus be used for emulating SceneScan when performing systems development, or for transmitting image data to SceneScan when using network image input.

The transmitted processing results always consist of two images. Usually these are the rectified left camera image and the computed left camera disparity map. If configured, however, SceneScan can also provide the raw camera images or the rectified images of both cameras (see Section 9.9).

Original and rectified camera images are typically transmitted with a bit-depth of 8 bits or 12 bits per pixel. The disparity map is always transmitted with a bit depth of 12 bits. Inside the library, the disparity map and any 12-bit images are inflated to 16 bits, to allow for more efficient processing.

The API provides three classes that can be used for receiving and transmitting image pairs:

- `ImageProtocol` is the most low-level interface. This class allows for the encoding and decoding of image pairs to / from network messages. You will have to handle all network communication yourself.

- `ImageTransfer` opens up a network socket for sending and receiving image pairs. This class is single-threaded and will thus block when receiving or transmitting data.

- `AsyncTransfer` allows for the asynchronous reception or transmission of image pairs. This class creates one or more threads that handle all network communication.

Detailed information on the usage of each class can be found in the available API documentation.

10.2 ImageTransfer Example

An example for using the class `ImageTransfer` in C++ to receive processing results over the network, and writing them to image files, is shown below. This source code file is part of the API source code release. Please refer to the API documentation for further information on using `ImageTransfer` and for examples in Python.

---

[3]http://nerian.com/support/software/

```cpp
#include <visiontransfer/deviceenumeration.h>
#include <visiontransfer/imagetransfer.h>
#include <visiontransfer/imagepair.h>
#include <iostream>
#include <exception>
#include <stdio.h>

#ifdef _MSC_VER
// Visual studio does not come with snprintf
#define snprintf _snprintf_s
#endif

using namespace visiontransfer;

int main() {
  try {
    // Search for SceneScan devices
    DeviceEnumeration deviceEnum;
    DeviceEnumeration::DeviceList devices = deviceEnum.discoverDevices();
    if(devices.size() == 0) {
      std::cout << "No devices discovered!" << std::endl;
      return -1;
    }

    // Print devices
    std::cout << "Discovered devices:" << std::endl;
    for(unsigned int i = 0; i< devices.size(); i++) {
      std::cout << devices[i].toString() << std::endl;
    }
    std::cout << std::endl;

    // Create an image transfer object that receives data from
    // the first detected SceneScan
    ImageTransfer imageTransfer(devices[0]);

    // Receive 100 images
    for(int i=0; i<100; i++) {
      std::cout << "Receiving image " << i << std::endl;

      // Receive image
      ImagePair imagePair;
      while(!imageTransfer.receiveImagePair(imagePair)) {
        // Keep on trying until reception is successful
      }

      // Write both images one after another
      for(int imageNumber = 0; imageNumber <=1; imageNumber++) {
        // Create PGM file
        char fileName[100];
        snprintf(fileName, sizeof(fileName), "image%03d_%d.pgm", i,
          imageNumber);

        imagePair.writePgmFile(imageNumber, fileName);
```

```
      }
    }
  } catch(const std::exception& ex) {
    std::cerr << "Exception occurred: " << ex.what() << std::endl;
  }

  return 0;
}
```

10.3 AsyncTransfer Example

An example for using the class `AsyncTransfer` in C++ to receive processing results over the network, and writing them to image files, is shown below. This source code file is part of the API source code release. Please refer to the API documentation for further information on using `AsyncTransfer` and for examples in Python.

```
#include <visiontransfer/deviceenumeration.h>
#include <visiontransfer/asynctransfer.h>
#include <visiontransfer/imagepair.h>
#include <iostream>
#include <exception>
#include <stdio.h>

#ifdef _MSC_VER
// Visual studio does not come with snprintf
#define snprintf _snprintf_s
#endif

using namespace visiontransfer;

int main() {
  try {
    // Search for SceneScan devices
    DeviceEnumeration deviceEnum;
    DeviceEnumeration::DeviceList devices = deviceEnum.discoverDevices();
    if(devices.size() == 0) {
      std::cout << "No devices discovered!" << std::endl;
      return -1;
    }

    // Print devices
    std::cout << "Discovered devices:" << std::endl;
    for(unsigned int i = 0; i< devices.size(); i++) {
      std::cout << devices[i].toString() << std::endl;
    }
    std::cout << std::endl;

    // Create an image transfer object that receives data from
    // the first detected SceneScan
    AsyncTransfer asyncTransfer(devices[0]);
```

```cpp
    // Receive 100 images
    for(int i=0; i<100; i++) {
      std::cout << "Receiving image " << i << std::endl;

      // Receive image
      ImagePair imagePair;
      while(!asyncTransfer.collectReceivedImagePair(imagePair,
        0.1 /*timeout*/)) {
        // Keep on trying until reception is successful
      }

      // Write both images one after another
      for(int imageNumber = 0; imageNumber <=1; imageNumber++) {
        // Create PGM file
        char fileName[100];
        snprintf(fileName, sizeof(fileName), "image%03d_%d.pgm", i,
          imageNumber);

        imagePair.writePgmFile(imageNumber, fileName);
      }
    }
  } catch(const std::exception& ex) {
    std::cerr << "Exception occurred: " << ex.what() << std::endl;
  }

  return 0;
}
```

10.4   3D Reconstruction

As described in Section 8.2, the disparity map can be transformed into a set of 3D points. This requires knowledge of the disparity-to-depth mapping matrix Q (see Section 8.2), which is transmitted by SceneScan along with each disparity map.

An optimized implementation of the required transformation, which uses the SSE or AVX instruction sets, is provided by the API through the class `Reconstruct3D`. This class converts a disparity map into a map of 3D point coordinates. Please see the API documentation for further details.

11   Supplied Software

11.1   NVCom

The available source code or binary software release also includes the NVCom client application, which is shown in Figure 31. When compiling this application yourself, please make sure that you have the libraries OpenCV and Qt installed. NVCom provides the following features:
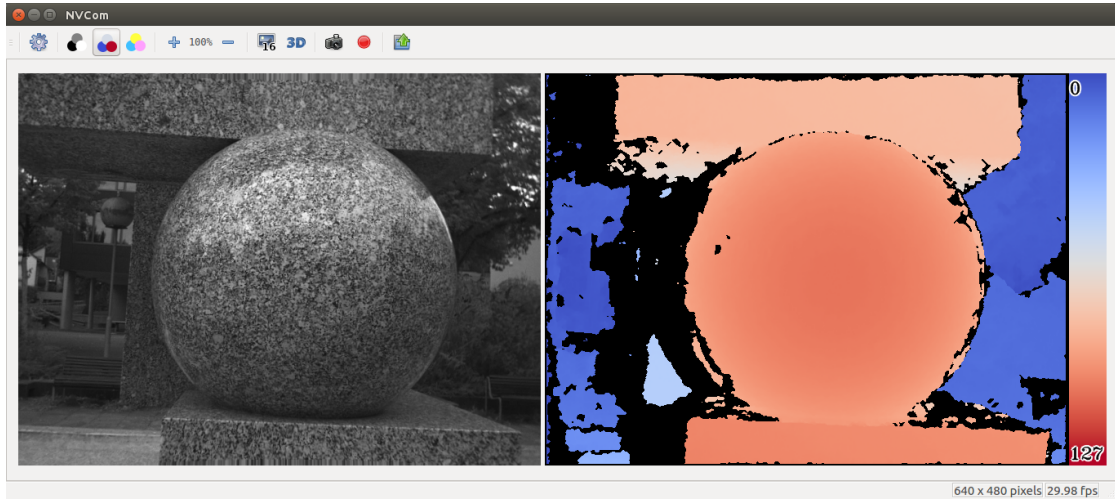
47

Figure 31: Screenshot of NVCom application.

- Discover SceneScan devices, view their status, and access their setup.

- Receive and display images and disparity maps from SceneScan.

- Perform color-coding of disparity maps.

- Write received data to files as images or 3D point clouds.

- Transmit image pairs to SceneScan.

NVCom comes with a GUI that provides access to all important functions. More advanced features are available through the command line options, which are listed in Table 3. The command line options can also be used for automating data recording or playback.

Unless NVCom is run in non-graphical mode, it opens a GUI window that displays the received image pair. The currently displayed image pair can be written to disk by pressing the *enter* key or by clicking the camera icon in the toolbar. When pressing the *space* key or clicking the recording icon, all subsequent images will be saved. When closing NVCom it will save its current settings, which will be automatically re-loaded when NVCom is launched the next time.

11.2   GenICam GenTL Producer

11.2.1   Installation

The available software release further includes a software module that complies to the GenICam GenTL standard. The GenTL standard specifies a generic transport layer interface for accessing cameras and other imaging devices. According to the GenICam naming convention, a GenTL producer is a software driver that provides access to an imaging device through the GenTL interface.

Table 3: Available command line options for NvCom.

`--help`	Displays information about the available command line options.
`-c` *color scheme*	Select a color coding scheme for received disparity maps (0 = no color, 1 = red / blue, 2 = rainbow).
`-s` *directory*	Send the images from the given directory. If not specified, only image reception will be active.
`-f` *fps*	Limits the frame rate to the given value. This option can be used for controlling the frame rate when transmitting images.
`-d`	Disable image reception. Use this option if you want to transmit images only.
`-n`	Non-graphical mode. No GUI window will be opened.
`-w` *directory*	Immediately start writing all received images to the given directory.
`-3` *max. distance*	Whenever capturing a frame, also write a 3D point cloud with distances up to the given maximum (0 = off).
`-b` *on/off*	Write point clouds in binary rather than text format.
`-h` *hostname*	Use the given remote hostname for communication.
`-p` *port*	Use the given remote port number for communication.
`-t` *on/off*	Activate / deactivate the usage of TCP as underlying network protocol. If TCP is not selected, UDP is used instead.
`-r` *on/off*	Output the disparity map (right image) as raw 16-bit image without bit-depth conversion.
`-i` *0/1*	Only displays the left (0) or right (1) image.
`-T`	Print frame timestamps to the console.
`-z` *percent*	Set zoom factor to the given percentage.
`-F`	Run in fullscreen mode.

A GenTL consumer, on the other hand, is any software that uses one or more GenTL producers through this interface. The supplied software module represents a GenTL producer and can be used with any application software that acts as a consumer. This allows for the ready integration of SceneScan into existing machine vision software suites like e.g. HALCON.

Depending on the version that you downloaded, the producer is provided either as a binary or as source code. If you choose the source code release, the producer will be built along with the other software components. The produced / downloaded binary is named `nerian-gentl.cti`. In order to be found by a consumer, this file has to be placed in a directory that is in the GenTL search path. The search path is specified through the following two environment variables:

`GENICAM_GENTL32_PATH`:   Search path for 32-bit GenTL producers.
`GENICAM_GENTL64_PATH`:   Search path for 64-bit GenTL producers.

The binary Windows installer automatically configures these environment variables. When building the source code release, please configure the environment variables manually.

11.2.2 Virtual Devices

Once the search path has been set, the producer is ready to be used by a consumer. For each SceneScan the producer provides five *virtual devices*, which each deliver one part of the obtained data. These virtual devices are named as follows:

/left Provides the left camera image that is transmitted by SceneScan. In the default configuration, this is the image after rectification has been applied. The image is encoded with 8 or 12 bits per pixel (Mono8 or Mono12).

/right Provides the image of the right camera. The right camera image is only transmitted if SceneScan is configured in pass through or rectify mode (see Section 9.9). The image is encoded with 8 or 12 bits per pixel (Mono8 or Mono12).

/disparity Provides the disparity map that is transmitted by SceneScan. This data is not available if SceneScan is configured in pass through or rectify mode. The disparity map is transmitted with a non-packed 12 bits per pixel encoding (Mono12).

/pointcloud Provides a transformation of the disparity map into a 3D point cloud (see Section 8.2). Each point is represented by three 32-bit floating point numbers that encode an x-, y- and z-coordinate (Coord3D_ABC32f).

/ This virtual device provides a multi-part data stream which contains all the data that is available through the other devices. In the default configuration, this device provides the left camera image, the disparity map and the 3D point cloud.

The virtual devices */left*, */right* and */disparity* deliver the unprocessed data that is received from SceneScan. The data obtained through the */pointcloud* device is computed by the producer from the received disparity map. This is done by multiplying the disparity map with the disparity-to-depth mapping matrix *q* (see Section 8.2), which is transmitted by SceneScan along with each image pair. Invalid disparities are set to the minimum disparity and thus result in points with very large distances.

It is recommended to use the multi-part virtual device / when more than one type of data is required. This will guarantee that all data acquisition is synchronized. When requiring only one type of input data, then using the dedicated virtual devices is the most efficient option.

11.2.3 Device IDs

All device IDs that are assigned by the producer are URLs and consist of the following components:

$$\boxed{protocol}\ :\!/\!/\ \boxed{address}\ /\ \boxed{virtual\ device}$$

The *protocol* component identifies the underlying transport protocol that shall be used for communication. The following values are possible:

udp: Use the connection-less UDP transport protocol for communication.

tcp: Use the connection oriented TCP transport protocol for communication.

The *virtual device* shall be set to one of the device names that have been listed in the previous section. Some examples for valid device IDs are:

```
udp://192.168.10.10/pointcloud
tcp://192.168.10.100/left
```

11.3   ROS Node

For integrating SceneScan with the Robot Operating System (ROS), there exists an official ROS node. This node is called `nerian_stereo` and can be found in the official ROS package repository. The node publishes the computed disparity map and the corresponding 3D point cloud as ROS topics. Furthermore, it can publish camera calibration information.

To install this node from the ROS package servers on a Ubuntu Linux system, please use the following commands:

```
> sudo apt-get update
> sudo apt-get install ros-'rosversion -d'-nerian-stereo
```

Detailed information on this node can be found on the corresponding ROS wiki page[4].

12   Support

If you require support or if you have other inquiries that are related to this product, please contact:

Nerian Vision GmbH
Zettachring 2
70567 Stuttgart
Germany
Phone: +49 711 2195 9414
E-mail: service@nerian.com

Website: www.nerian.com

13   Warranty Information

The device is provided with a 2-year warranty according to German federal law (BGB). Warranty is lost if:

- the housing is opened by others than official Nerian Vision Technologies service staff.

- the firmware is modified or replaced, except for official firmware updates.

In case of warranty please contact our support staff.

14   Open Source Information

SceneScan's firmware contains code from the open source libraries and applications listed in Table 4. Source code for these software components and the wording of the respective software licenses can be obtained from the open source information website[5]. Some of these components may contain code from other open source projects, which may not be listed here. For a definitive list, please consult the respective source packages.

The following organizations and individuals have contributed to the various open source components:

Free Software Foundation Inc., Emmanuel Pacaud, EMVA and contributors, The Android

---

[4]http://wiki.ros.org/nerian_stereo
[5]http://nerian.com/support/resources/scenescan-open-source/

Table 4: Open source components.

Name	Version	License(s)
Aravis	0.6.4 patched	GNU LGPL 2.0
GenApi reference implementation	3.1.0	GenICam License
libwebsockets	2.2	GNU LGPL 2.1
ntp	4.2.8p10	BSD License
		MIT License
OpenCV	3.2.0	BSD License
		libpng License
		JasPer License 2.0
OpenSSL	1.1.1d	BSD License
PetaLinux	2019.2	Various
PHP	7.3.7	PHP License
PTPd	git (July 18, 2019)	BSD

Open Source Project, Red Hat Incorporated, University of California, Berkeley, David M. Gay, Christopher G. Demetriou, Royal Institute of Technology, Alexey Zelkin, Andrey A. Chernov, FreeBSD, S.L. Moshier, Citrus Project, Todd C. Miller, DJ Delorie, Intel Corporation, Henry Spencer, Mike Barcroft, Konstantin Chuguev, Artem Bityuckiy, IBM, Sony, Toshiba, Alex Tatmanjants, M. Warner Losh, Andrey A. Chernov, Daniel Eischen, Jon Beniston, ARM Ltd, CodeSourcery Inc, MIPS Technologies Inc, Intel Corporation, Willow Garage Inc., NVIDIA Corporation, Advanced Micro Devices Inc., OpenCV Foundation, Itseez Inc., The Independent JPEG Group, elibThomas G. Lane, Guido Vollbeding, Simon-Pierre Cadieux, Eric S. Raymond, Mans Rullgard, Cosmin Truta, Gilles Vollant, James Yu, Tom Lane, Glenn Randers-Pehrson, Willem van Schaik, John Bowler, Kevin Bracey, Sam Bushell, Magnus Holmgren, Greg Roelofs, Tom Tanner, Andreas Dilger, Dave Martindale, Guy Eric Schalnat, Paul Schmidt, Tim Wegner, Sam Leffler, Silicon Graphics, Inc. Industrial Light & Magic, University of Delaware, Martin Burnicki, Harlan Stenn, Danny Mayer, The PHP Group Perseus Telecom, George N. Neville-Neil, Jan Breuer, George V. Neville-Neil, Steven Kreuzer, Wojciech Owczarek, Alexandre Van Kempen, Inaqui Delgado, Rick Ratzel, National Instruments, Gael Mace, OpenSSL Software Services, Inc., OpenSSL Software Foundation, Inc., Andy Polyakov, Ben Laurie, Ben Kaduk, Bernd Edlinger, Bodo Möller, David Benjamin, Emilia Käsper, Eric Young, Geoff Thorpe, Holger Reif, Kurt Roeckx, Lutz Jänicke, Mark J. Cox, Matt Caswell, Matthias St. Pierre, Nils Larsch, Paul Dale, Paul C. Sutton, Ralf S. Engelschall, Rich Salz, Richard Levitte, Stephen Henson, Steve Marquess, Tim Hudson, Ulf Möller, Viktor Dukhovni

All authors contributing to packages included in PetaLinux. Please obtain the full list from www.xilinx.com/petalinux.

If you believe that your name should be included in this list, then please let us know.

Revision History

Revision	Date	Author(s)	Description
v1.9	July 1, 2020	KS	New simple and advanced settings. New Acquisition configuration page.
v1.8	January 20, 2020	RYT, KS	ROI selection; trigger pulse width cycling
v1.7	November 20, 2019	RYT, KS	DHCP support; improved subpixel optimization.
v1.6	August 14, 2019	KS	Updated specifications for firmware 4.0.0; Add Karmin3 information.
v1.5.1	July 23, 2019	KS	Minor corrections.
v1.5	April 1, 2019	KS	Support for Bayer pattern inputs.
v1.4	November 30, 2018	KS	Added new support for color image processing; added Baumer cameras. Minor wording improvements.
v1.3	June 7, 2018	KS	Changes for vision software release 6.0.0 / firmware release 2.0.0
v1.2	March 1, 2018	KS	Added PTP synchronization
v1.1	February 1, 2018	JH, KS	Auto exposure; frame rate table; supported pixel formats.
v1.0	September 27, 2017	KS	Initial revision