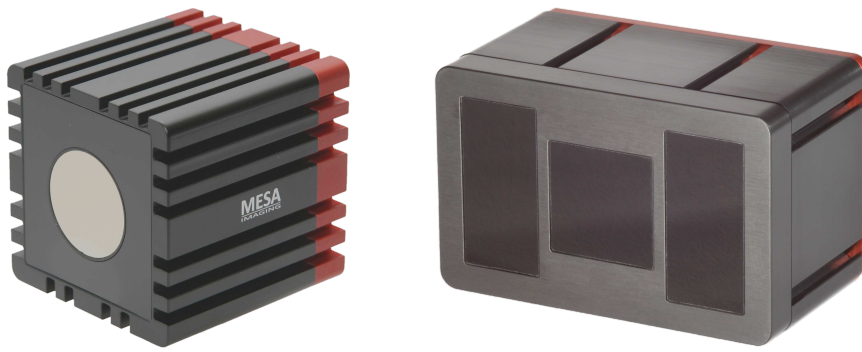


# SR4000/SR4500 User Manual



**MESA**  
IMAGING

Version 3.0

**Contents**

<b>1</b>	<b>About this document</b>	<b>5</b>
1.1	Goal of this document . . . . .	5
1.2	Target group . . . . .	5
1.3	Symbols used in this document . . . . .	6
<b>2</b>	<b>Safe usage of the camera</b>	<b>7</b>
2.1	Authorized personnel . . . . .	7
2.2	Correct use of the camera . . . . .	7
2.3	Camera hardware integrity . . . . .	7
2.4	Safety notes and protective measures . . . . .	7
2.4.1	Electrical installation of the camera . . . . .	8
2.4.2	Eye safety . . . . .	9
<b>3</b>	<b>Environmental protection</b>	<b>10</b>
<b>4</b>	<b>General recommendations</b>	<b>11</b>
4.1	Heat sink requirements . . . . .	11
4.2	Power requirements . . . . .	11
4.3	Cleaning the camera . . . . .	11
<b>5</b>	<b>Package contents</b>	<b>13</b>
5.1	Shipped components . . . . .	13
5.1.1	SR4000 . . . . .	13
5.1.2	SR4500 . . . . .	13
5.2	Description . . . . .	14
5.2.1	SR4000 . . . . .	14
5.2.2	SR4500 . . . . .	15
<b>6</b>	<b>How Time Of Flight works</b>	<b>17</b>
6.1	Distance estimation and 3D measurement . . . . .	17
<b>7</b>	<b>Measurement performance</b>	<b>19</b>
7.1	Definition of absolute accuracy and repeatability . . . . .	19
7.2	Factors that influence absolute accuracy and repeatability . . . . .	19
7.2.1	Absolute accuracy . . . . .	19
7.3	Specified absolute Accuracy and repeatability . . . . .	19
7.3.1	About the reflectivity of different materials . . . . .	20
7.3.2	Method to determine the reflectivity of targets . . . . .	20
7.4	Movement Artifacts . . . . .	21
7.5	Setting the optimal integration time . . . . .	21
<b>8</b>	<b>Physical setup</b>	<b>23</b>
8.1	Environmental conditions . . . . .	23
8.1.1	Temperature . . . . .	23

8.1.2	Ambient light and sunlight . . . . .	23
8.2	Avoiding multiple reflections . . . . .	23
8.3	Non-ambiguity range . . . . .	24
8.3.1	SR4000 . . . . .	25
8.3.2	SR4500 . . . . .	25
8.4	Multiple cameras . . . . .	26
8.4.1	SR4000 . . . . .	26
8.4.2	SR4500 . . . . .	26
<b>9</b>	<b>Data output of the camera</b>	<b>27</b>
9.1	Distance data . . . . .	27
9.2	XYZ Cartesian coordinates . . . . .	28
9.3	Amplitude data . . . . .	28
9.4	Confidence map . . . . .	29
<b>10</b>	<b>Image acquisition modes</b>	<b>30</b>
10.1	Continuous acquisition mode . . . . .	30
10.2	Triggered acquisition modes . . . . .	32
10.2.1	Software trigger mode . . . . .	32
10.2.2	SR4000 hardware trigger mode . . . . .	33
<b>11</b>	<b>Application Programming Interface (API) overview</b>	<b>35</b>
11.1	Interfacing functions . . . . .	35
11.1.1	Opening and closing . . . . .	35
11.1.2	Image capture . . . . .	37
11.1.3	Control parameters for image capture . . . . .	38
11.2	Parameters affecting measurement quality . . . . .	38
11.2.1	Integration Time . . . . .	39
11.2.2	Amplitude threshold . . . . .	39
11.2.3	Modulation frequency . . . . .	40
11.2.4	Auto exposure . . . . .	40
11.3	Filters and auxiliary functions . . . . .	41
<b>12</b>	<b>Driver installation and setup</b>	<b>42</b>
12.1	Installing driver, demo and sample software . . . . .	43
12.1.1	Windows . . . . .	43
12.1.2	Linux . . . . .	43
12.2	Uninstalling and reinstalling the driver software . . . . .	43
12.3	Connecting a USB camera . . . . .	45
12.3.1	Windows . . . . .	45
12.4	Connecting an Ethernet camera: using DHCP or static IP address . . . . .	45
12.4.1	Connecting to an SR4000 using DHCP . . . . .	45
12.4.2	Setting a static IP address . . . . .	45
12.4.3	Reverting to DHCP . . . . .	46
12.5	Silent install in Windows . . . . .	46
<b>13</b>	<b>Software applications</b>	<b>47</b>
13.1	Using the SwissRangerSampleGUI Application . . . . .	47
13.1.1	Distance image . . . . .	48
13.1.2	Grayscale image . . . . .	49

13.1.3	Scaling of display color . . . . .	49
13.1.4	Important adjustments . . . . .	49
13.1.5	Acquisition options . . . . .	49
13.2	Visualizing and capturing 3D data using the SR_3D_View Application . . . . .	49
13.2.1	Display options . . . . .	51
13.2.2	Filters . . . . .	51
13.2.3	Streaming . . . . .	52
13.2.4	Further controls . . . . .	52
13.3	Sample Code . . . . .	53
13.4	Matlab Interface . . . . .	53
13.5	C# Interface . . . . .	53
<b>14</b>	<b>Help and Support</b>	<b>54</b>
<b>15</b>	<b>Electrical power and trigger connection</b>	<b>55</b>
15.0.1	SR4000 . . . . .	55
15.0.2	SR4500 . . . . .	55
<b>16</b>	<b>Ratings and conformance</b>	<b>58</b>

## 1 About this document

This manual allows you to operate with the SR4000/SR4500 type time-of-flight (TOF) cameras in an efficient way.

*Please read the complete manual carefully before working with the camera.*

In this document the following designations:

- SR4000/SR4500
- Camera
- Sensor

will all be used, and will always refer to the SR4000/SR4500 TOF camera. There are some differences between the SR4000 and SR4500:

- The SR4500 has more light power and is more robust against background light.
- The SR4500 contains two new features (non-ambiguity range extension and multi-camera mode) that the SR4000 does not offer.
- The SR4500 is IP67 rated.
- The SR4500 does not have any hardware trigger functionality.

Some of these differences are addressed by split subsections in this document.

### 1.1 Goal of this document

This manual is designed to

- instruct customers about basic functions of the SR4000/SR4500,
- help customers to use the SR4000/SR4500 in an optimal way,
- and to address aspects of the safe electrical installation, configuration, maintenance and usage of the SR4000/SR4500 TOF cameras.

### 1.2 Target group

This manual is addressed to technicians/engineers with sufficient skills to understand it and to apply all relevant measures/steps to ensure safe usage of the SR4000/SR4500 according to the manual instructions and according to the safety regulation in place in the specific country where the camera is used.

### 1.3 Symbols used in this document

**✓ Important**

*A note “important” indicates important information that the reader should understand/follow. Such a note is in most of the cases related to optimizing the measurement performance of the camera.*

**x WARNING**

*A note “warning” indicates an actual or potential hazard. Warnings are designed to help you to prevent accidents. Read carefully and follow the warning notices.*

## 2 Safe usage of the camera

This chapter addresses topics related to the safety of the operation of the camera. Every camera user must read this chapter carefully before working with the SR4000/SR4500.

### 2.1 Authorized personnel

**✓ Important**

*The SR4000/SR4500 must be installed and used by adequately qualified personnel only!*

Personnel is only considered as qualified if it has sufficient skills to understand and to apply all relevant measures/steps to ensure safe usage of the SR4000/SR4500 according to the manual instructions and according to the safety regulation in place in the specific country where the camera is used.

### 2.2 Correct use of the camera

The SR4000/SR4500 is a measurement device that allows capturing 3D data of IR light reflective objects in the surrounding scene. The camera can be used by itself or can be integrated into a larger system that uses the 3D data delivered by the camera.

**✓ Important**

*In case of any other camera usage other than stated above any claims against MESA Imaging AG will be rendered void.*

The camera must be used in environments that are adapted to the camera enclosure rating.

### 2.3 Camera hardware integrity

By opening the camera or by modifying the camera hardware the declaration of CE conformity, the safety, as well as the warranty of the camera will be rendered void. MESA Imaging AG cannot be made faulty/responsible of any financial loss or injuries to users from cameras that have been opened or modified.

**✓ Important**

*Under no circumstances is it allowed to open the camera or to make any changes to the camera hardware!*

### 2.4 Safety notes and protective measures

Please observe the following items in order to ensure the correct and safe use of the SR4000/SR4500:

- The notices in this document must be observed.
- When operating the SR4000/SR4500, the national, local and statutory rules and regulations must be observed. In particular following regulations must be observed:
  - the work safety regulations/safety rules
  - other relevant health and safety regulations
- Manufacturers and operators of the system on which the SR4000/SR4500 is installed are responsible for obtaining and observing all applicable safety regulations and rules.
- The operating instructions must be made available to the operator of the system where the SR4000/SR4500 is integrated. The operator of the system is to be instructed in the use of the device by specialist personnel and must be instructed to read the operating instructions.
- The SR4000/SR4500 is not a device that is certified for safety applications

**✘ WARNING**

*Please keep in mind the safety notes and protective measures when working with the camera.*

### 2.4.1 Electrical installation of the camera

Please, observe the following items:

- Only adequately qualified personnel is allowed to install the camera
- Only the standard power supply provided by MESA may be used with the camera.
- Another power supply might only be installed by an adequately qualified personnel. In this case, following applies:
  - The power supply isolation from the ground voltage must be redundant (i.e. double isolated power supply).
  - The power supply must be conform to chapter 2.5 of EN60950-1:2006 to ensure fire safety.
  - Wire cross-sections and their correct fuse protection as per the applicable standards must be implemented.
  - All notices from chapter 15 of this document have to be applied.
  - MESA cannot be made responsible for any injuries resulting from a faulty power supply design.
- The user must ensure that no significant electrical grounding potential difference might appear between the camera and the network/power supply. This is in particular true for cameras connected to Ethernet networks that span over different physical buildings.
- Follow the current safety regulations when working on electrical systems.
- Do not open the housing.

**✘ WARNING**

*The notes on electrical installation of the camera must be followed.*



### 2.4.2 Eye safety

The sensor emits light actively. Therefore, rules and laws related to active light emitting technical devices apply. The following statements hold for the SR4000/SR4500:

- The SR4000/SR4500 operates with LEDs emitting in the 850 nm range.
- The SR4000/SR4500 is certified LED Class 1 according to EN 60825-1:2002 (eye safe).

**X WARNING**

*Pay attention to the laser safety regulations as per EN 60825-1 (latest version). Eye safety is only proven for cameras that are used in normal operation conditions. In particular, cameras should never be supplied with voltages exceeding those specified as in chapter 15.*

### **3 Environmental protection**

The camera contains electronic components. When you dispose it pay attention to the following recommendations:

- Always dispose devices that are not serviceable or repairable in compliance with local/national rules and regulations on waste disposal.
- Dispose of all electronic assemblies as hazardous waste. The electronic assemblies are straightforward to dismantle.

## 4 General recommendations

### 4.1 Heat sink requirements

#### ✓ Important

Provide for suitable heat sinking as explained in this section.

When the SR4000/SR4500 is in use, especially when it will be operating for long periods, it is important to ensure that there is always adequate heat sinking. The requirements for heat sinking depend on the environment and application. The essential requirement is that sufficient heat is drawn away from the camera to ensure that the temperature of the camera housing does not exceed 50 °C. Heat may be drawn away by mounting the camera to a larger thermal mass (provided that the mounting is thermally conductive), by using forced airflow (especially if there is any form of enclosure), or by using a heat sink with “fins” if a static system is required.

Longer Integration Times (see chapter 11 for an explanation of the integration time parameter) cause the LEDs to be active for longer periods, resulting in more heat. To reduce the amount of heat generated, and power consumed, it may help to use *triggered* rather than the *continuous* mode (see chapter 10).

### 4.2 Power requirements

#### ✓ Important

For the camera to operate with the precision as described in the specifications, the SR4000 must be powered with +12 V DC, Min -2%, Max +10% and the SR4500 must be powered with +24 V DC, Min -10%, Max +10%.

If the minimal voltage is not supplied, the camera will have less precise measurements. If the camera is supplied with a higher voltage, the exceeding energy will be dissipated through heating of the camera, which will also influence the camera's precision.

On demand MESA delivers a closed housing power supply that complies with those requirements, but that is not of industrial grade (especially in respect to lifetime under demanding conditions). For applications where reliability over time is a key requirement, an industrial grade power supply must be used to power the camera. MESA does not supply those devices, but can suggest some suited devices/suppliers if needed (send a request to <mailto:support@mesa-imaging.ch>).

### 4.3 Cleaning the camera

Together, the optical filter and the LED cover (see chapter 5) form a flat front face which may be cleaned by wiping with a lint-free cloth, dampened with isopropyl alcohol if necessary, or with optical wipes. If the front face is dirty with particles that may be abrasive, care must be taken not to scratch the surface.

**✓ Important**

*Clean the front end with appropriate utilities, e.g. as explained above.*

## 5 Package contents

### 5.1 Shipped components

The shipped components contain the packaging material, an end-test qualification report and the hardware components. These are described in the following subsections for each camera type separately.

#### 5.1.1 SR4000

A standard SR4000 package contains:

1. SR4000 camera (USB or Ethernet)
2. Software and documentation Installation CD

Optionally, it may also contain the following items :

3. Communications cable (USB or Ethernet)
4. Power supply unit
5. Power cable

The regular and optional items for a SR4000 USB camera are shown in figure 5.1.



Figure 5.1: SR4000 package contents

The Ethernet camera packages contain an Ethernet cable instead of the USB cable (see ❸ in figure 5.1). Additionally, a trigger cable (not shown in figure 5.1) may be part of the package if appointed.

#### 5.1.2 SR4500

A standard SR4500 package contains:

1. SR4500 camera

2. Software and documentation Installation CD

Optionally, it may also contain the following items :

3. Mounting slot nuts and tripod adapter
4. Power supply unit
5. Ethernet cable
6. Power cable

The regular and optional items are shown in figure 5.2.



- 1 SR4500 camera
- 2 Software and documentation CD
- 3 Mounting slot nuts and tripod adapter
- 4 Power supply unit
- 5 Ethernet cable
- 6 Power cable

Figure 5.2: SR4500 package contents

## 5.2 Description

### 5.2.1 SR4000

The front view of the SR4000 is shown in figure 5.3.



Figure 5.3: Front view SR4000

The following components are shown in figure 5.3:

- Illumination cover. This protects the LEDs while allowing their light to be transmitted.
- Optical Filter. This allows only light of wavelengths near that of the illumination LEDs to pass into the camera lens.

Views of the SR4000 from the back and below are shown in figure 5.4.

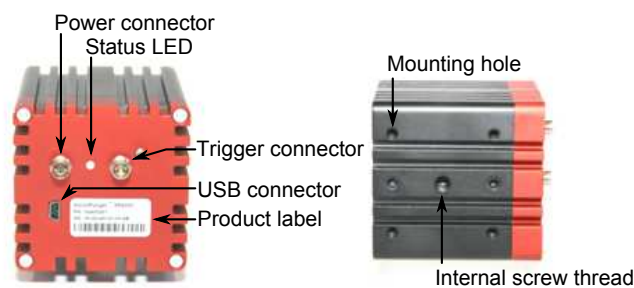


Figure 5.4: Back view showing connectors and view from below showing mounting holes

The following components are shown in figure 5.4:

- Power connector. For supply of 12 V DC at up to 1 A to the camera.
- Trigger connector. For external hardware triggering of image acquisition.
- Data connector (USB in this case). For connection to PC.
- Status LED. Flashing slow = power on. Flashing fast = acquiring images.
- Mounting holes and internal screw thread. Precise and robust mounting of the camera can be made through four M4 holes and two additional 4H7 holes. In addition, it is possible to screw the camera onto standard tripods (usually used for photo-cameras).

### 5.2.2 SR4500

The front view of the SR4500 is shown in figure 5.5.

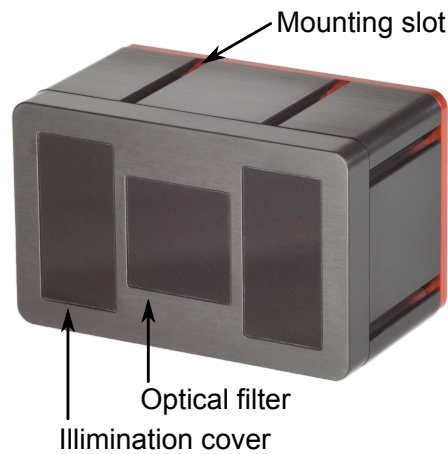


Figure 5.5: Front view SR4500

The following components are shown in figure 5.5:

- Illumination cover. This protects the LEDs while allowing their light to be transmitted. Below the right hand side illumination cover is a status LED that flashes on each image frame during acquisition and with 1 Hz if the camera is operational but no images are taken.
- Optical Filter. This allows only light of wavelengths near that of the illumination LEDs to pass into the camera lens.
- Mounting slot. This slot is reserved for the slot nuts that are also included in the package. They allow flexible mounting configuration while providing mechanical stability.

A view of the SR4500 from the back is shown in figure 5.6.

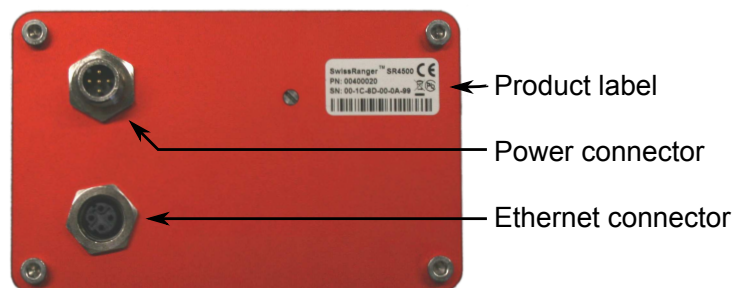


Figure 5.6: Back view showing connectors

The following components are shown in figure 5.6:

- Product label. This label contains the product type number and the serial number. The serial number is needed if the camera needs to be returned for repair.
- Power connector. For supply of 24 V DC at up to 1 A to the camera.
- Ethernet connector. For connection to PC.



## 6 How Time Of Flight works

The distance measurement capability of the SR4000 is based on the Time of Flight (TOF) principle. See e.g. <http://en.wikipedia.org/wiki/Time-of-flight> for a description of this principle.

In Time of Flight systems, the time taken for light to travel from an active illumination source to IR light reflective objects in the field of view and back to the sensor is measured. Given the speed of light  $c$ , the distance can be determined directly from this round trip time. To achieve the time of flight measurement the SR4000 modulates its illumination LEDs, and the CCD/CMOS imaging sensor measures the *phase* of the returned modulated signal at each pixel. The distance at each pixel is determined as a fraction of the one full cycle of the modulated signal, where the distance corresponding to one full cycle is given by

$$D = \frac{c}{2f} \quad (6.1)$$

where  $c$  is the speed of light and  $f$  is the modulation frequency. At a modulation frequency of 30 MHz, this distance is 4.997 m at a speed of light 299792458 m/s. Therefore, 30 MHz cameras are those with the 5 m range in the product specifications and 15 MHz cameras are those with the 10 m range. The analog electrical signals are converted into digital values in an Analog to Digital conversion process, from which a 14-bit value is calculated. This “raw” output of the camera refers to the spherical distance measure between the pixel on the camera chip and the corresponding region in the environment. From the single pixel values a 176 by 144 pixel depth map is computed.

### 6.1 Distance estimation and 3D measurement

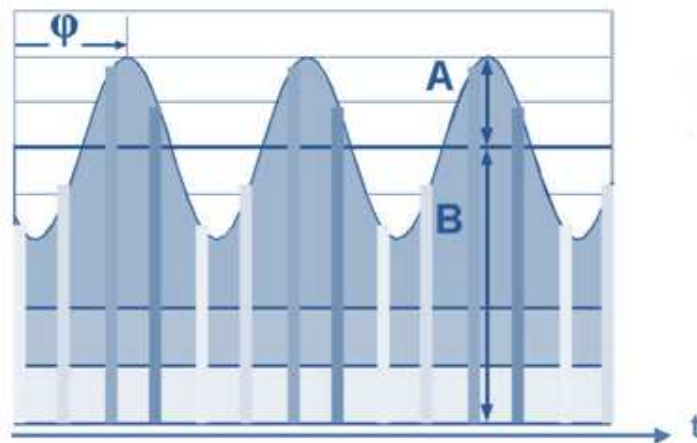


Figure 6.1: Time of Flight sampling of returned modulated signal

The LEDs in the front of the sensor emit modulated light pulses. The modulation signal can be thought of a sinusoidal signal as shown in figure 6.1. As shown in figure 6.1 by the vertical lines the reflected signal is sampled four times in each cycle, at 4 period phase shifts, i.e. 90° phase angle. Signal B is the mean of the total light incident on the sensor: background plus modulated signal, and A is the Amplitude of just the modulate signal. The phase  $\phi$  (phi) is calculated from the four samples

to produce the distance measurement. The amplitude  $A$  may be used as a measure of quality of the distance measurement, or to generate a gray scale image.

One burst of the modulated signal is sent out for each image and multiple phase shifts are integrated during this time. The lengths of the integration process can be adjusted by the user by means of a parameter called Integration Time (IT) (see also section 7.5 and chapter 11).

The raw 14-bit values of the spherical distances range from 0 to 16382 (0 to 0x3FFE in hex). Higher values indicate saturation. A single bit increment corresponds to  $0.305\text{ mm}$  (rounded) if  $2^{14}$  for a  $5\text{ m}$  camera and  $0.61\text{ mm}$  (rounded) for a  $10\text{ m}$  camera. These values can be obtained by dividing the full phase lengths by  $2^{14}$ .

The real spherical distance in  $\text{mm}$  can in be obtained by multiplying the 14-bit phase measure with the factor of  $0.305\text{ mm}$  or  $0.61\text{ mm}$  respectively. From this distance together with the intrinsic parameters of the camera  $x$ ,  $y$ , and  $z$  coordinates per pixel are computed. This is done inside the camera so that it delivers full 3D coordinate vectors for each of the 176 times 144 pixels.

## 7 Measurement performance

### 7.1 Definition of absolute accuracy and repeatability

To characterize the precision of a camera a statistical approach is used. For an object at a given (known) distance, repeating the distance measurement with the camera will result in a distribution of measured values centered on a mean value.

**Absolute accuracy** is the difference between the mean measured value of the distribution and the real value and represents the maximum systematic error on the distance measurement. This value is particularly relevant for the precision on absolute distance measurement.

**Repeatability** is characterized by the spread ( $1\sigma$ ) of the measurement around the mean value and is relevant to characterize the precision on a single measurement. This value gives an indication on the noise of a given measurement.

### 7.2 Factors that influence absolute accuracy and repeatability

#### Repeatability

Repeatability (noise) of distance measurements depends mainly on signal amplitude and background illumination. Signal amplitude in turn depends on object distance and object reflectivity (see section 7.3.1). The optimal repeatability is achieved when the integration time is set to give the greatest Amplitude without reaching saturation.

The repeatability (noise) of distance measurements is often improved using some form of temporal or spatial averaging.

#### 7.2.1 Absolute accuracy

Absolute accuracy on the other hand is independent of distance, and reflectivity. The absolute accuracy is achieved internally in the SR4000 through a combination of elements including pixel and electronics architectures, an optical feedback loop, and temperature compensations. In nominal operation an absolute accuracy of  $< 1\text{ cm}$  is achievable (see specification or data sheet for more information).

### 7.3 Specified absolute Accuracy and repeatability

Every camera type has a data sheet that gives more details about the guaranteed absolute accuracy and repeatability. Please refer to this document for the specifications of your camera.

As described before, the reflectivity of the measured object has a large influence on the repeatability of the measurement. This is the reason why it is important to understand how the reflectivity of an object is defined and how it can be measured.

**7.3.1 About the reflectivity of different materials**

The measurement characteristics of the camera highly depend on the reflection characteristics of the scene and objects that are measured. To describe the reflectivity of objects, two aspects of the reflected light must be considered:

1. The proportion of the back scattered light (energy) compared to the incoming light (energy).
2. The angular distribution of the back scattered intensities (directed reflection versus diffuse reflection)

For diffusely reflecting materials (mat and dull surfaces), the reflectivity coefficient has values between 0% (all incoming light is absorbed or transmitted) and 100% (such as for a white sheet of paper). The reference of 100% is given by the case of a perfect Lambertian reflector, where 100% of the light is back-scattered with an intensity distribution that is independent of the observation angle.

For directed reflecting materials, such as glossy surfaces, Retro-reflectors, or in the extreme case mirrors, the reflection coefficient might be  $\geq 100$  for specific angles at which the light is directly reflected. Camera measurements are not optimal for such directed reflections as the image might saturate at the angles where all the light is reflected directly into the sensor. Another problem is when the reflected beam points away from the camera, preventing the sensor from capturing enough signal intensity to deliver stable measurements. See table 7.1 for examples of different materials and their reflectivity.

Material	Reflectivity	Material	Reflectivity
Kodak gray card (bright)	107%	Rough clean wood palette	25%
White paper	80 – 100%	Smooth concrete	25%
White masonry	85%	Kodak gray card (dark)	33%
Newspaper	70%	Black rubber tire	2%
PVC (grey)	40%		

Table 7.1: typical reflectivity values for diffusely reflecting materials at a wavelength of 850 nm and at 90° incidence

**7.3.2 Method to determine the reflectivity of targets**

In order to determine the reflectivity of a surface, an object or a target, a simple method consists in placing a target of known reflectivity directly next to this surface, and to measure under the same angle the relative intensity of the surface compared to the reflectivity of the reference target.

As reference target one can use Kodak gray cards, as they are quasi-Lambertian reflectors for the visible light, are of constant quality and are worldwide available (Kodak R27, Ref 1527795). Nevertheless, note that the Kodak gray cards are optimized for visible light, and not for the 850 nm wave length at which we are operating. Table 7.2 gives a comparison between reflectivity coefficient of Kodak paper for the visible and the infrared (850 nm) domain.

	For visible light (as indicated by Kodak)	For infrared light
<b>Bright</b>	90%	107.1%
<b>Dark</b>	18%	32.9%

Table 7.2: Reflectivity coefficient of Kodak gray cards at 90° incidence

### 7.4 Movement Artifacts

Each of the 4 phase measurement samples (see figure 6.1 on page 17) are taken as a separate exposure. This means that in order to obtain a phase/distance measurement, four consecutive exposures must be performed. If an object in the scene moves during these exposures, errors may be introduced into the measurements. The noise introduced due to movement is random noise at the edges of the object moved.

### 7.5 Setting the optimal integration time

In order to set the optimal integration time one has to compromise between 3 aspects:

1. The needed acquisition speed
2. The needed measurement quality
3. The type of measured objects

The diagram in figure 7.1 shows the interdependencies of these aspects.

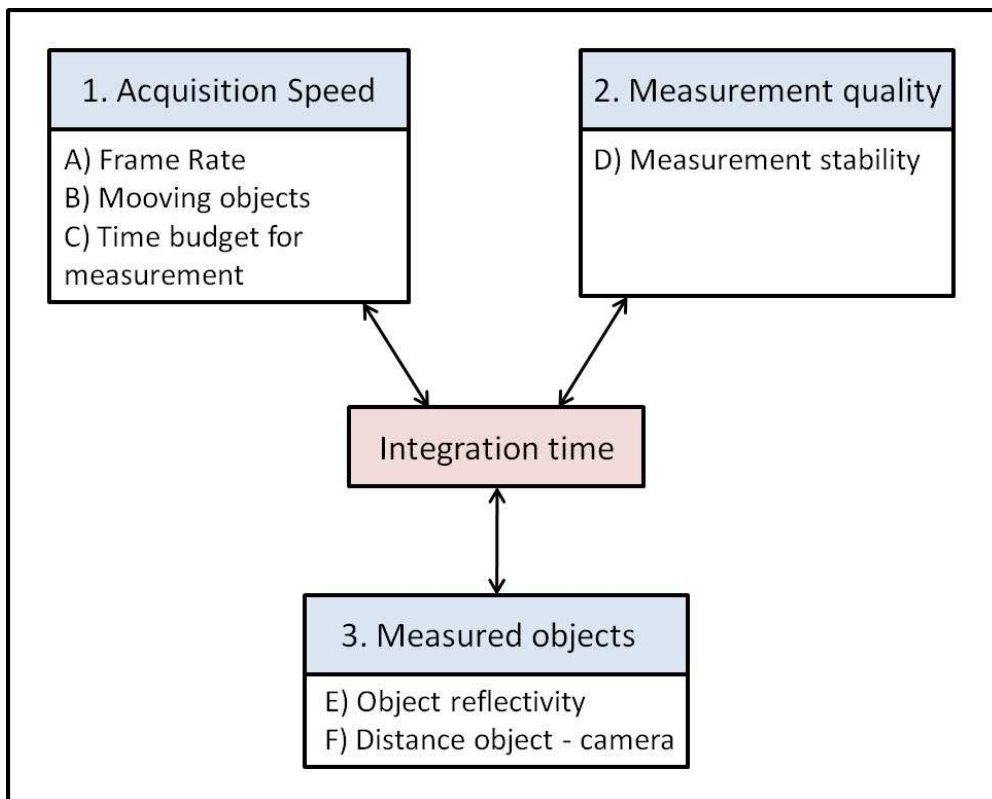


Figure 7.1: Diagram showing interdependencies of different parameters

In the following enumeration the diagram is explained in more detail:

1. Acquisition speed

The frame rate at which the images are acquired is inversely proportional to the integration time. Setting a smaller integration time will result in

- A Higher frame rate

- B Less movement artifacts from moving objects
  - C Minimizing the time budget needed for the overall measurement (e.g. measurements that involve image processing on a series of images)
2. Measurement quality
- D The noise on the measurement ( $1\sigma$ ) is inversely proportional to the integration time: longer integration times allow to capture a larger amount of reflected light, which results in lower noise (also described as higher repeatability).
3. Type of measured objects
- E The amount of light that is reflected from the measured objects is proportional to their reflectivity (in the infrared domain). Thus, to reach the same measurement quality (repeatability), a higher integration time needs to be set for low-reflective objects than for high reflective objects.
  - F The light intensity decreases with the square of the traveled distance. Thus to reach the same measurement quality (repeatability), objects located further away need a higher integration time than objects located closer to the camera.

## 8 Physical setup

### 8.1 Environmental conditions

The camera should be mounted securely in free airflow for cooling, free from vibration, with care taken to ensure the conditions as specified in this chapter in order to ensure precise and stable measurements.

#### 8.1.1 Temperature

**✓ Important**

*The camera must always be mounted to allow sufficient heat to be drawn away from the camera.*

The requirement is to ensure that the camera's housing temperature does not exceed 50 °C. Heat may be drawn away by following measures:

- Mount the camera to a larger thermal mass (provided that the mounting is thermally conductive)
- Use forced airflow (especially if there is any form of enclosure).
- Use a heat sink with fins.
- Using longer integration times causes the camera to generate more heat than using short integration times. In the case that continuous measurement is not needed, heat generation may be minimized by using triggered acquisition (in chapter 10 the different acquisition modes are explained).

#### 8.1.2 Ambient light and sunlight

**✓ Important**

*Direct sunlight will affect the performance. The camera should not be used in direct sunlight.*

Indirect sunlight that shines through doors and windows in indoor environments is usually not a problem.

### 8.2 Avoiding multiple reflections

**✓ Important**

*Avoid setups with multiple light paths.*

The total distance traveled by the light is about twice the distance from the camera to the object. However, it is possible that objects may be arranged in the scene such that light takes a less direct path than this.

For example, the light may be reflected by a first object before being reflected by the measured object and finally return to the camera sensor. In this situation the light travels by the direct and also indirect

paths (hence the term multipath). The apparent distance is then a weighted average of the path distances, weighted by the strength of signal returned via each path. The end result is that distance measurements are over-estimated.

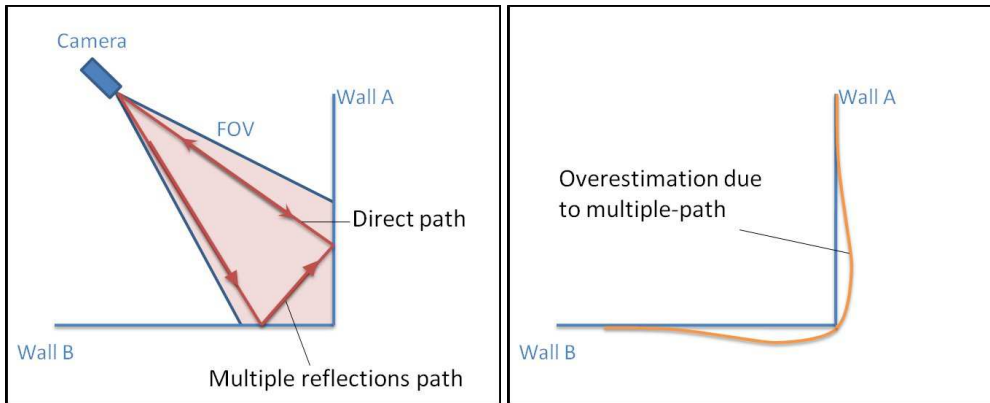


Figure 8.1: Multiple path reflections on a concave scene - in that case a corner between two walls

A common situation of multipath appears when measuring objects that have concave structures. A good example is when measuring a scene with a corner between two walls. In that case, the region of the wall right next to the corner will be seen further away than it is in reality (see figure 8.1, orange line). This is due to the fact that a large portion of the IR light is reflected on the neighbor wall first (wall B), then on the measured wall (wall A) into the camera. Maximum overestimation is in the region where multiple reflection path are both, maximum in number and in intensity. This explains the shape of the measured values, as shown in orange on figure 8.1 (right hand side).

In order to avoid as much as possible multipath problems, the camera should be placed at the greatest distance possible from other interfering objects from which light might be reflected. Ideally, the Field of View of the camera should not include any objects that have not to be measured. Two frequent setups that might cause problems and that have to be avoided are cameras close to walls or mounted on top of tables (see figure 8.2).

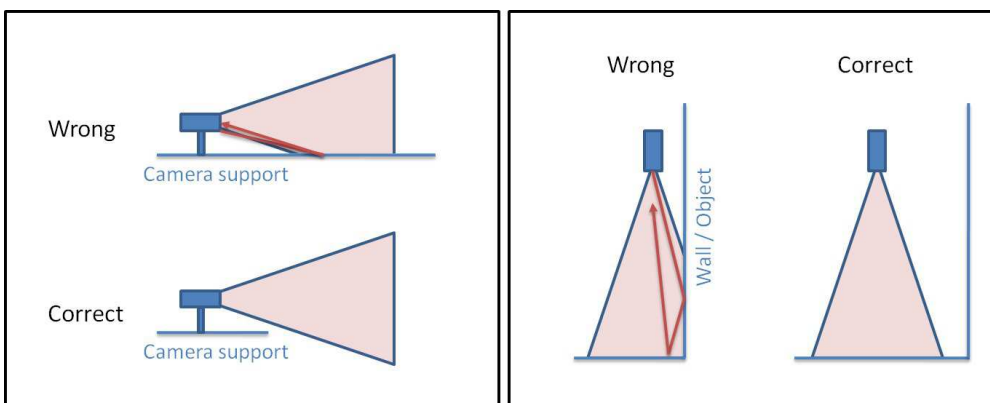


Figure 8.2: Positioning the camera to avoid reflections. The FOV of the camera (in light red) should not include any objects that has not to be measured and that could result in multiple path reflections.

### 8.3 Non-ambiguity range



8.3.1 SR4000

Measurements are subject to a so called “ambiguity” or a “back-folding” phenomenon that is due to the periodicity of the signal that is used for the distance measurement. If objects could be present in the scene at distances which differ by more than the distance corresponding to a full modulation period  $D$  (see figure ?? on page 17), the measurement of their position is ambiguous: it could be at  $x$  or at  $x + D$ , or even  $x + 2D$  etc ( $x + nD, n \in \{1, 2, \dots\}$ ). For this reason the full phase distance is referred to as the “non-ambiguity range” that is either  $0 - 5\text{ m}$  or  $0 - 10\text{ m}$  depending on the camera type.

It follows that in the case of a camera with a measurement range of  $5\text{ m}$ , any object that is situated further than  $5\text{ m}$  and whose intensity is still high enough to be detected by the camera will be folded back into the “non ambiguity range” (e.g.: an object at  $7\text{ m}$  will be measured at  $2\text{ m}$ , one at  $9\text{ m}$  at  $4\text{ m}$  for a  $5\text{ m}$  camera and one at  $13\text{ m}$  at  $3\text{ m}$  for a  $10\text{ m}$  camera). Similarly, for a camera with a range of  $10\text{ m}$ , all detectable objects situated at  $12\text{ m}$  will be folded back at  $2\text{ m}$ , etc.

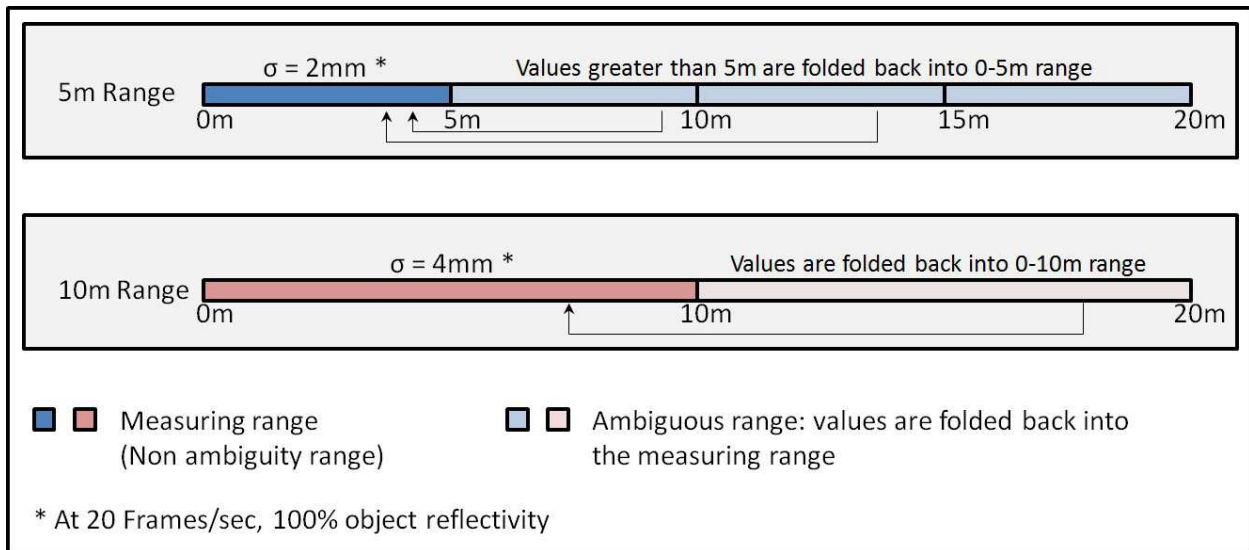


Figure 8.3: Illustration of the Non-ambiguity range

In the field, some applications are limited in range by the geometry of the environment (e.g. a wall in the background at less than  $5\text{ m}$ ). In that case a  $5\text{ m}$  camera will deliver reliable results, with no artifacts due to objects folded back into the measurement range (the same holds for scene with a limited range of  $10\text{ m}$  using a  $10\text{ m}$  camera).

When the background is not limited, the folding back of far and bright objects will cause problems. The solution is to filter those values out by setting an amplitude threshold. This filtering works best using a  $10\text{ m}$  camera, as practice has shown that most objects situated further than  $10\text{ m}$  meters have low intensities.

8.3.2 SR4500

The SR4500 provides a new feature called non-ambiguity range extension (NARE). This feature overcomes the ambiguity problem as described above in the SR4000 subsection. The feature is activated on the SR4500 by default. The corrective computations carried out by this mode need two sequential images. For applications where faster frame rates are more important than avoiding ambiguity errors this feature can be turned off. See chapter 11 for information about how this can be done. The NARE requires an amplitude threshold to be set (see chapter 11 for information about how to set this

parameter) in order to work correctly.

**✓ Important**

*Always use an appropriate amplitude threshold when working with the NARE feature!*

## 8.4 Multiple cameras

### 8.4.1 SR4000

It is possible to connect multiple USB or Ethernet cameras to the same host computer. However, in any situation where multiple SR4000 cameras are operating in the same area the illumination of the camera may interfere with other cameras. For the SR4000 there are possibilities to avoid this interference:

- Sequence the exposures of each camera using triggered acquisition mode. A custom software application is required to sequence the cameras. The disadvantages of this approach are that the frame rate is significantly diminished for each camera included in the system, and that the acquisitions from each camera are not simultaneous. The latter may be important when movement is present in the scene.
- Different modulation frequencies may be used in each camera to enable them to work together with no interference. The SR4000 supports several different modulation frequencies:
  - For the 5 m range cameras
    - \* 29 MHz (5.17 m range)
    - \* 30 MHz (5.00 m range)
    - \* 31 MHz (4.84 m range)
  - For the 10 m range cameras
    - \* 14.5 MHz (10.34 m range)
    - \* 15 MHz (10.00 m range)
    - \* 15.5 MHz (9.68 m range)

### 8.4.2 SR4500

The SR4500 has a built-in feature that avoids mutual disturbances and therefore, no special care needs to be taken about this problem.

## 9 Data output of the camera

This section describes the data output of the camera. The camera outputs the following data per pixel:

- distance data (phase measure)
- $(x, y, z)$  coordinate data
- amplitudes (similar to gray-level data)
- and a confidence map (an estimate of reliability).

All these data are described in the following sections.

On programming level the data may be accessed using the C/C++ software API of the driver or one of the interfaces to Matlab or Python. On application level sample and viewer programs are available on the software and documentation CD or via download from <http://www.mesa-imaging.ch/drivers.php>. These programs are described in chapter 13 (software) and can be used to visualize the camera's data outputs.

### 9.1 Distance data

The distance data is given by the phase angle  $\phi$  as described previously in chapter 6. This angle is directly proportional to the distance that light travels from the LEDs to the object and back. The phase angles for all pixels are available with the API function `SR_GetImage()` as a pointer to a  $176 \cdot 144 = 25344$  long array of unsigned short values. The data is arranged in this array as a 144 sequence of rows of size 176. The phase angles are output from the SR4000 as 16 bit values where only the upper 14 bits (D13..D0, see table 9.1, below) are used. In table 9.1 the format of the pixel data is shown.

D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	ID1	ID0
-----	-----	-----	-----	----	----	----	----	----	----	----	----	----	----	-----	-----

Table 9.1: Arrangement of the bits composing the distance values.

The two least significant bits (ID1 and ID0 in table 9.1) are reserved. The distance value  $d$  of a pixel is given by

$$d = \frac{\phi \cdot d_{max}}{2^{14}} \quad (9.1)$$

where  $\phi$  is the 14 bit pixel value encoded by D13 to D0 and  $d_{max}$  the maximal full-phase distance of the camera on the camera type (see chapter 6).

The hex value 0x0000 (upper 14 bits) corresponds to the origin of an intrinsic coordinate system and 0x3FFF corresponds to the full-phase radial distance.

#### ✓ Important

*The last two values 0x3FFE and 0x3FFF do not represent distances but instead indicate saturation.*

## 9.2 XYZ Cartesian coordinates

The software driver provides a coordinate transform function which converts the raw 14 bit radial distances as explained in the previous section to Cartesian coordinates, expressed in meters. The calculations are performed by the driver on the host CPU. This transformation includes a correction which compensates for the radial distortion of the optics. The coordinate system used here is “Right-Handed”, with z-coordinate increasing along the optical axis away from the camera, y-coordinate increasing vertically upwards and x-coordinate increasing horizontally to the left, all from the point of view of the camera (or someone standing behind it). Figure 9.1 shows the camera’s output coordinate system.

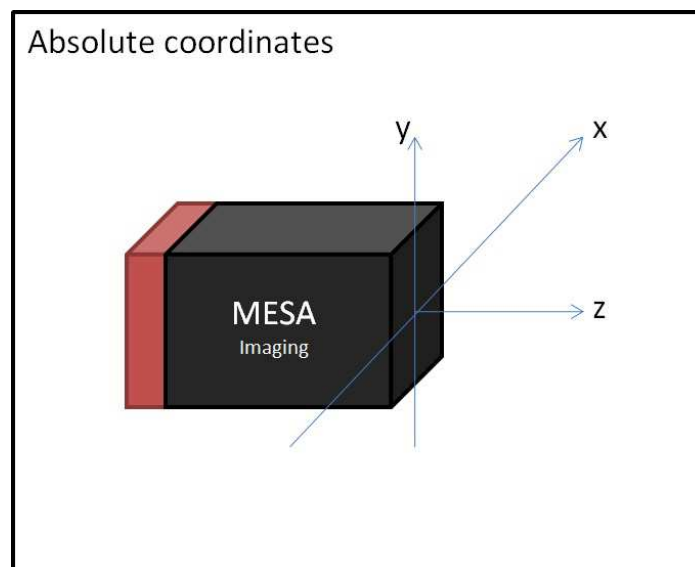


Figure 9.1:  $(x, y, z)$  as delivered by the camera is given in this coordinate system.

The origin of the coordinate system  $(0, 0, 0)$  is at the intersection of the optical axis with the front face of the camera. Saturated pixels transform to the value  $(0, 0, 0)$  (saturation is explained below in section 9.3). More information about the visualization of the camera outputs are given in chapter 13 (software) and the software interfacing is described chapter 11 (API description)

### ✓ Important

*The coordinate system used by the camera is not the same as usually used in the literature on perspective camera projection models. These systems lie in the center of lens (about 11 mm from the front plate into the housing) and the x and y are rotated around z positively about  $\pi/2$ . If you want to retrieve the exact system for your camera including all **projective camera parameters** then you can use the code snippet presented in our forum contact <http://forum.mesa-imaging.ch/viewtopic.php?f=33&t=169> or contact <mailto:support@mesa-imaging.ch>*

## 9.3 Amplitude data

Amplitude data, (quantity A in figure 6.1 on page 17), is output as an array of 16 bit words, arranged like the distance values described in section 9.1.

Again only the 14 most significant bits are used. The largest 14 bit value of 0x3FFF indicates saturation. Values from 0x000 to 0x3FFE are proportional to the received amplitudes. Bits ID1 and ID0 are reserved.

This arrangement is shown in table 9.2 on the next page.

A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	ID1	ID0
-----	-----	-----	-----	----	----	----	----	----	----	----	----	----	----	-----	-----

Table 9.2: Arrangement of the bits composing the amplitude values.

The raw amplitude data of the SR4000 are more intense in the image center and decrease towards the image borders. By default, the amplitude is converted into a value which is independent of distance and position in the image array, using the following factors:

1. A factor proportional to the square of the measured distance, scaled to equal 1 at distance of half of the full-phase distance.
2. A factor which corrects for the drop in strength of the illumination away from the center of the field of view. This factor equals 1 at the center, and increases with radial distance from the center.

Due to an improved illumination this effect is much smaller for the SR4500. It follows that the distance data of the SR4500 in the edges of the image are more stable compared to the SR4000.

#### 9.4 Confidence map

The confidence map is generated in the driver of the host PC using a combination of distance and amplitude measurements and their temporal variations. It represents a measure of probability or “confidence” of how correct the distance measurement is expected to be. Low confidence is typically due to low reflected signal or movement in the scene. The Confidence Map can be used to

- select regions containing measurements of high quality
- reject measurements of low quality
- obtain a confidence measure for a measurement derived from a combination of many pixels
- estimate an output reliability in recognition algorithms

Confidence map data is output from the SR4000/SR4500 as an array of 16 bit words. There is one confidence value per pixel. The Confidence Map has a range of 0-0xFFFF, with greater values representing higher confidence.

## 10 Image acquisition modes

For each image, four samples of the reflected light are acquired and read out by the sensor. The recombination of the information from the four samples is done in the calculation phase. The output of the calculation are range values per pixel. This data is finally transferred to an external device (computer).

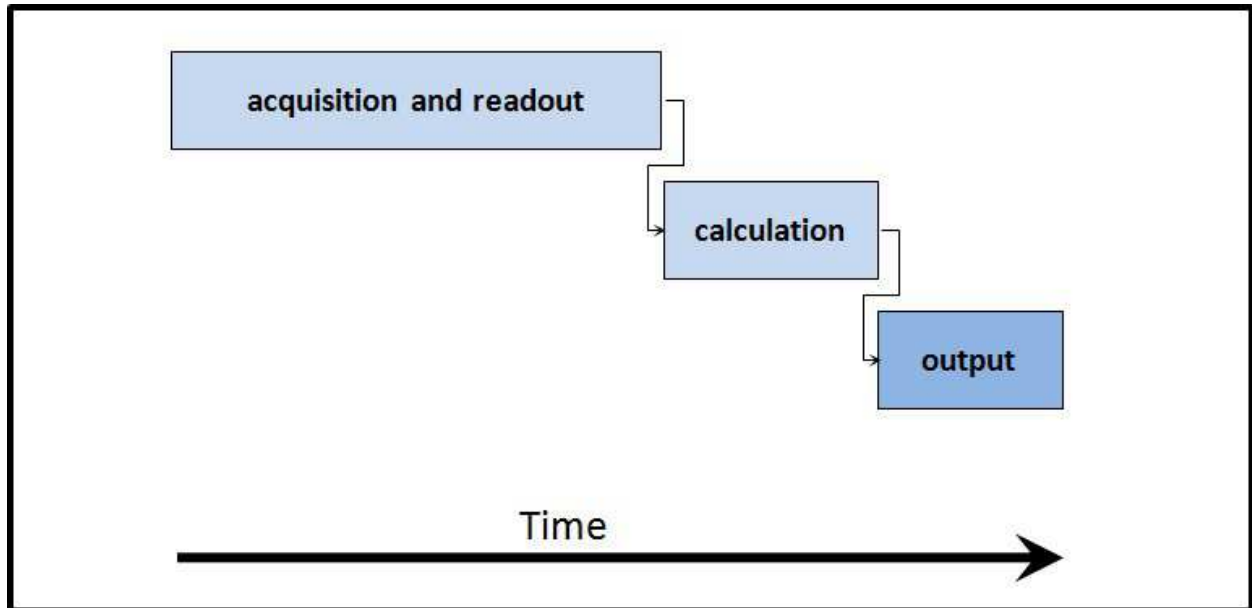


Figure 10.1: Schematic view of the image acquisition process

Two modes of image acquisition are supported:

1. Continuous acquisition mode
2. Triggered acquisition mode (also called single shot mode). In that mode, the acquisition may be triggered by either a software or a hardware trigger. The hardware trigger mode is only available for the SR4000. Two types of hardware trigger signal are available on the camera:
  - Trigger In: triggers the start of the acquisition and the data output
  - Trigger Out: signal that is emitted in order to synchronize external devices

The setting of the trigger modes is described in the API documentation or in chapter 11. In figure 10.2 on the next page an overview of the different trigger modes that the SR4000 supports is shown. The SR4500 supports the continuous acquisition mode and the single shot software trigger mode only.

### 10.1 Continuous acquisition mode

When software trigger mode is disabled, the camera captures images continuously (so called *continuous acquisition mode*). While acquisition and readout occurs for one image, the previous image is simultaneously processed in the internal FPGA. When an acquire command is received by the driver, the most

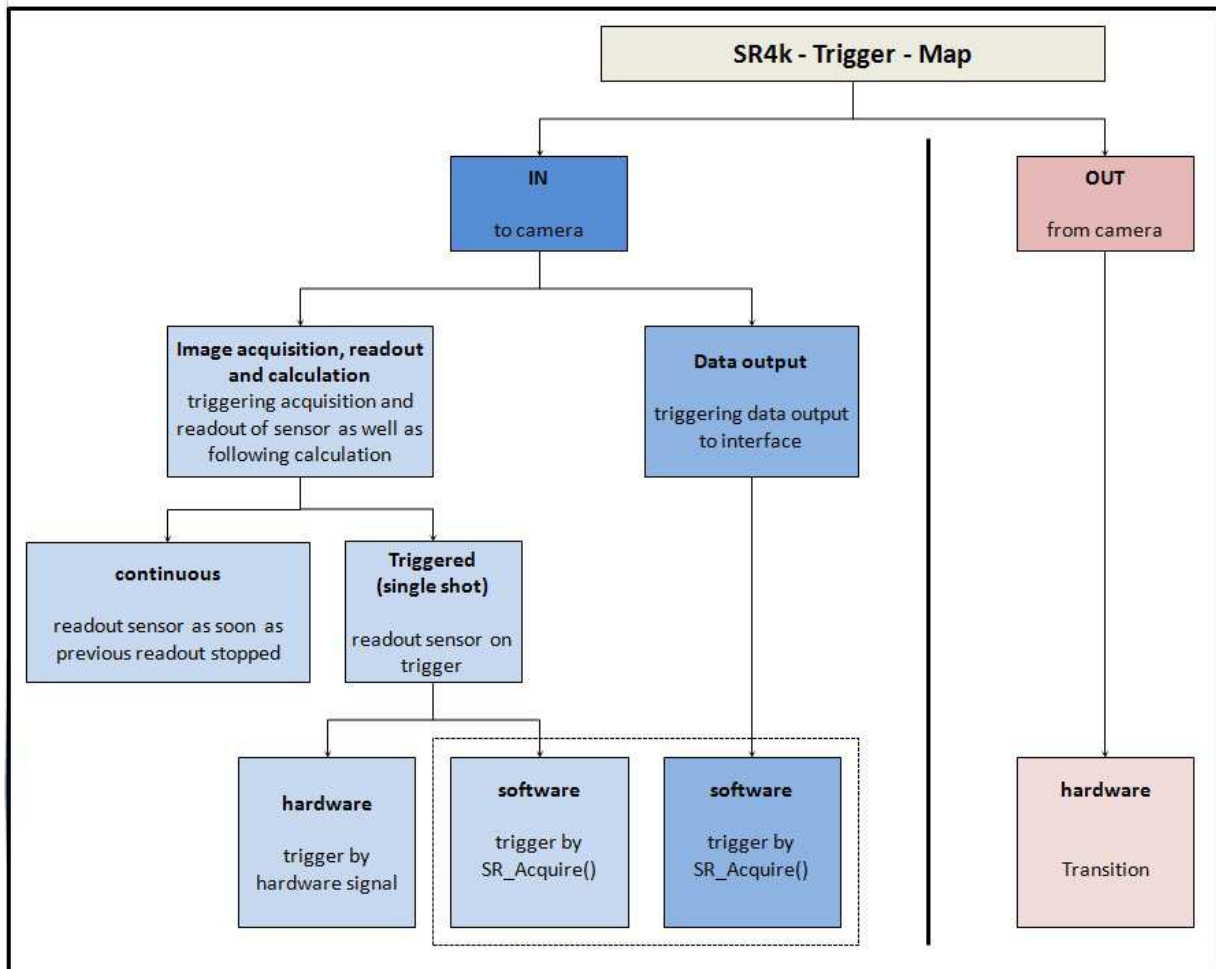


Figure 10.2: Trigger map of the SR4000. Trigger IN (blue) are a group of modes that are used to initiate an image capture. Trigger OUT (red) is an output signal from the camera that activates on the end of an acquisition cycle. See the related sections in the text for more details on the trigger signals. Note, that the SR4500 only supports the continuous acquisition mode and the single shot software trigger mode.

recent image whose processing has been completed by the FPGA is transferred to the host computer. In this way a high frame rate is achieved, but there is a latency in the output after completion of the acquisition of at least one frame. The continuous acquisition mode is not suited for synchronization with external devices, as the exact timing of the output of the image is not known externally.

The hardware trigger output consists of a pulse that starts at the beginning of each new acquisition. Every 4<sup>th</sup> pulse is 1.5 ms long, all others are 1ms long. The voltage range is 4.5 V to 5.0 V (see also chapter 15).

**✓ Important**

*In Continuous Mode, because of the 1-frame latency, any changes in mode, integration time or modulation frequency setting do not affect the first acquired frame after the setting change, since this frame would have been acquired before the setting change.*

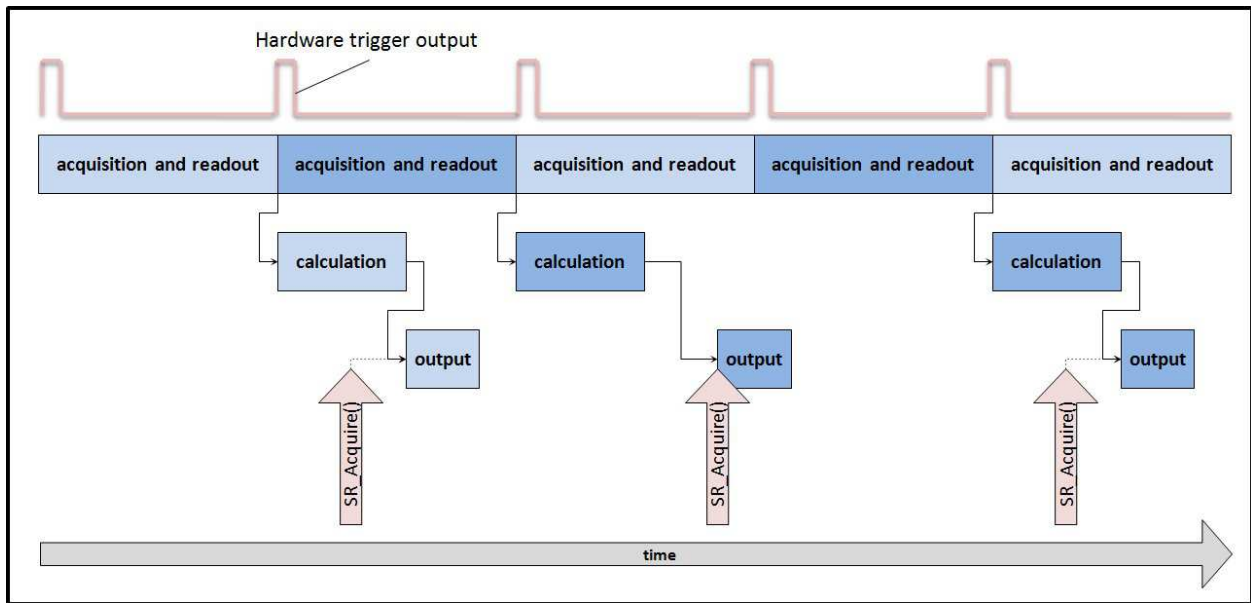


Figure 10.3: Schematic view of the continuous acquisition mode

**✓ Important**

*For very small integration times, the data transfer may not start until the beginning of the second subsequent (N+2) frame.*

## 10.2 Triggered acquisition modes

When software trigger mode is enabled, the camera does not capture an image until a trigger is received. The following two cases are possible:

1. Software trigger mode
2. Hardware trigger mode

These modes are described in the following subsections.

### 10.2.1 Software trigger mode

When software trigger mode is enabled, the camera waits for an acquire command. When the command is received the image capture starts. Once acquisition is complete the image is processed in the FPGA and then transferred to the host computer. On completion of the transfer the command completes. The acquired image is then available. A new acquisition will only start when the data transfer has been completed.

The hardware trigger output of the SR4000 is in active state for the duration of the acquisition and readout, and the falling transition of the hardware trigger output indicates that this phase is complete. The SR4500 does not support this feature.



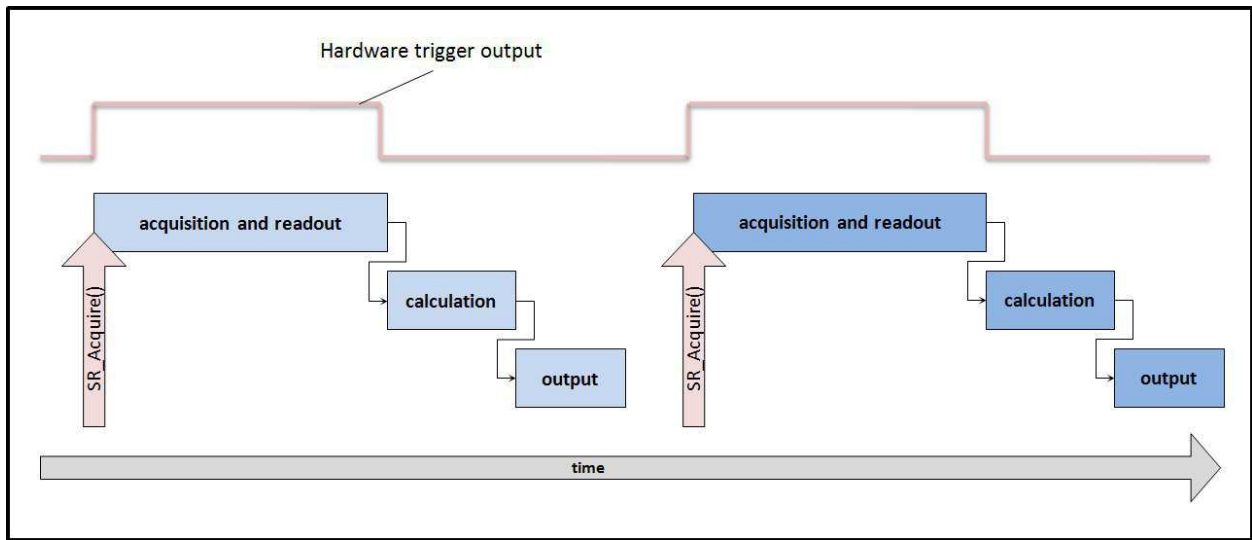


Figure 10.4: Schematic view of the software triggered mode

### 10.2.2 SR4000 hardware trigger mode

When hardware trigger mode is enabled in addition to the software trigger mode, the camera waits for an external hardware trigger. When the trigger signal is received the image acquisition starts. The Acquire command is then used to initiate transfer of the image to the host computer. When the command completes, the image is available. The hardware trigger output is in active state for the duration of the acquisition and readout, and the falling transition of the hardware trigger output indicates that this phase is complete.

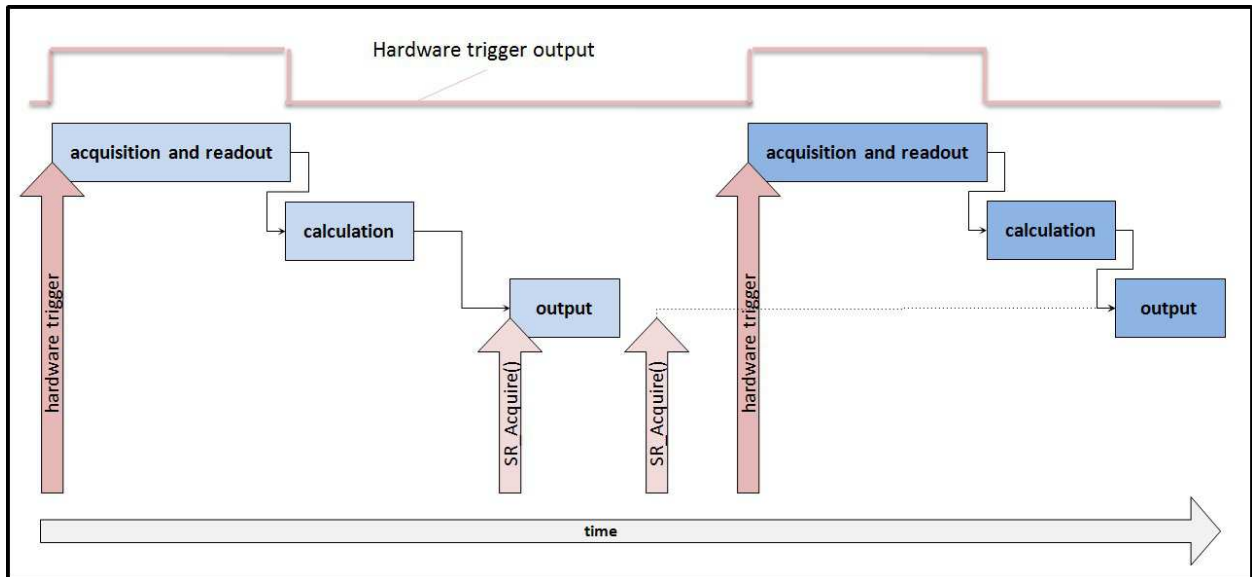


Figure 10.5: Schematic view of the hardware triggered mode

Hardware-triggered image capture may be parallelized subject to the following timing limitations:

- A second hardware trigger signal will trigger a new acquisition if it is received after the first image has completed acquisition and readout. The hardware trigger output may be used by an external system to detect when the integration is complete.

- If an acquire command is not received before the next image has completed integration then the first image is lost.
- If the previous image transfer has not completed before a second integration completes, the second image is lost.

## 11 Application Programming Interface (API) overview

The camera comes with a C/C++ application programming interface (API) including interfaces to Matlab and Python. The required data are on the CD shipped with the sensor or via download from <http://www.mesa-imaging.ch/drivers.php>. For two possibilities to access image data from C# see our forum:

- <http://forum.mesa-imaging.ch/viewtopic.php?f=33&t=146> or
- <http://forum.mesa-imaging.ch/viewtopic.php?f=33&t=170>.

In chapter 12 the installation of the software (drivers, documentation, and samples) is described.

The API is described with a comprehensive HTML documentation. Under Windows this is installed by default to:

```
C:\Program Files\MesaImaging\Swissranger\Swissranger.chm
```

and can be easily opened by following:

```
Start→Programs→MesaImaging→SwissRanger→SwissRanger API
```

from the Windows start panel. Linux user's can download the documentation from

```
http://www.mesa-imaging.ch/dlm.php?fname=./customer/driver/Swissranger-HTML-doc1.0.12.x.zip
```

The functions that are provided can be divided into three main groups:

- Functions that are related to interfacing with an application
- Functions that have impact on the image contents and measurement accuracy
- Filters and auxiliary function

In this chapter the first three sections are addressing the three groups as described above. The fourth section is devoted to some sample codes. Here the C/C++ functions are described. However, the same functionality is offered with the interfaces to Matlab and Python.

### 11.1 Interfacing functions

#### 11.1.1 Opening and closing

The camera is an instantiated object of the C/C++ type SRCAM defined in the file `libMesaSR.h`. API functions take a variable of this type as argument. In order to retrieve the camera's data it needs to be opened first, i.e. the communication via Ethernet or USB has to be initialized. The main opening functions are:

```
int SR_OpenDlg(SRCAM *srCam, int mode, HWND parent)
int SR_OpenUSB(SRCAM *srCam, unsigned int serialNumber)
int SR_OpenETH(SRCAM *srCam, const char *addr)
int SR_OpenFile(SRCAM *srCam, const char *filename)
```

All these functions open a camera object and are to be followed by `SR_Close()` to close the camera object. In the case of `int SR_OpenFile()` the camera object does not refer to a real camera but to a file stream that was recorded before. This enables for off-line development and testing. The opening

functions are described in the following paragraphs.

**Opening with a dialog** The function `SR_OpenDlg()` opens a camera that can be selected by the user in the selection menu. The following options are available and can be selected with the parameter mode:

1. try to open previous camera, if it failed error is returned, no selection dialog opened
2. always open selection dialog
3. try to open previous camera, if it failed, open selection dialog

In Windows the open dialog shows up as a selection window (see figure ?? on page ??). Ethernet and USB cameras are listed in this window. The Ethernet cameras must be in the same network as the host PC. You have to select a camera (or a camera file stream as described below) from the list with a left mouse click followed by a click on the **Connect** button.

The opening dialog can be accessed also with the `SwissRangerSampleGUI.exe` delivered with the Windows driver and with the demo program `SR_3D_View.exe` that can be downloaded from <http://www.mesa-imaging.ch/drivers.php>. If the IP address of the camera is known, then this address can be entered for direct opening. Under Linux the selection dialog is console based.

**Direct opening** The camera can be started directly (without any user feedback required) using `SR_OpenETH()` or `SR_OpenUSB()`. Which of these you have to use is depending on the actual product. If you have an Ethernet type SR4000 then `SR_OpenETH()` needs to be called with the IP address as parameter `const char *addr`. There are three ways an IP address can be provided to the camera:

- By default the IP address is requested from a DHCP server. In the upcoming selection dialog on `SR_OpenDlg()` (described above) the IP address is shown in the camera list. When there are more than one cameras found then you can identify the camera of interest by comparing the serial number (SN) printed on the backside of the housing to the MAC addresses in the selection dialog. These are equivalent.
- If the camera is booted up **without any Ethernet cable plugged in**, it enters a special mode where the so-called *fall back IP* is used, after about 20 seconds. In order to connect to the camera with this IP address you have to follow the recommendations given in section 12.4 on page 45.
- The fall back IP can be used to set a static IP address. In the case that a static IP address is set on the camera this address must be used to open the camera.

In case you have an USB type camera you can open with the function `SR_OpenUSB()`. It takes as argument the serial number of the camera. This number is printed on the back of the housing or it can be retrieved by the opening dialog described above. Also, the API function `SR_ReadSerial()` can be used to read the serial number.

**Opening from a file stream** Another option for opening is to open a SwissRanger stream (SRS). The function

```
int SR_OpenFile(SRCAM *srCam, const char *filename)
```

can be used to open a previously recorded stream. This stream behaves similar as a camera to serve for the following purposes:

- Development of recognition or general application programs without the availability of the actual camera.
- Reproducible camera tests on reference streams. The user can store “golden data”, e.g. detection

results in terms of image coordinates into a separate file. The SwissRanger file stream and the golden data file can be used for benchmarking in the development cycle.

- Simulation mode e.g. for unit tests.

The difference to a real camera is that all settings that affect the raw pixel data running on the camera cannot be reverted (see also section 11.2).

The API function

```
int SR_StreamToFile(SRCAM srCam, const char *filename, int mode)
```

can be used to record a stream. Also the sample and demo programs (see chapter 13) can be used for recording SRS streams. The function opens a file and streams received data from the camera to that file on each image acquisition. Again, it should be noted that those settings which affect the pixel data and that are run on the camera (see section 11.2) cannot be changed in future.

**✓ Important**

*When you use the streaming utility be aware that some parameters and settings as described in section 11.2 cannot be changed/reverted anymore. Thus a file stream is not a full logical replacement for a real camera.*

### 11.1.2 Image capture

The distance image and amplitude image (see chapter 9) are delivered by the functions

```
void* SR_GetImage(SRCAM srCam, unsigned char idx)
int SR_GetImageList(SRCAM srCam, ImgEntry **imgEntryArray)
```

in terms of pointers to unsigned short arrays. Note that the applications program does not “own” the data and thus it does not have to release them. The functions

```
int SR_CoordTrfUInt16(SRCAM srCam, short *x, short *y, unsigned short *z,
                    int pitchX, int pitchY, int pitchZ)
int SR_CoordTrfFlt(SRCAM srCam, float *x, float *y, float *z, |
                  int pitchX, int pitchY, int pitchZ) |
int SR_CoordTrfDbl(SRCAM srCam, double *x, double *y, double *z, |
                  int pitchX, int pitchY, int pitchZ) |
```

give pointers to the (x, y, z) coordinate vectors per pixel as described in chapter 9. These pointers refer to user allocated arrays.

**✓ Important**

*Be aware that the x, y, and z array pointers refer to user allocated memory. When these data are dynamically allocated then the user also has to release (free) them.*

The functions offer the output in different data types. The function SR\_CoordTrfUInt16() retrieves the original values (short for x and y, unsigned short for z) in millimeters. Furthermore, the function SR\_CoordTrfFlt() delivers 32 bit float and SR\_CoordTrfDbl() provides 64 bit double values in meters.

The pitchX, pitchY, and pitchZ arguments can be used for different arrangements of the data. This is best explained by example. The following code

```
int pitch = sizeof(short);
```

```
short* X = (short*)(malloc(rows * cols * pitch));
short* Y = (short*)(malloc(rows * cols * pitch));
short* Z = (short*)(malloc(rows * cols * pitch));
SR_CoordTrfUint16(cam, X, Y, (unsigned short*)Z), pitch, pitch, pitch)
```

will give you three separate arrays where the rows of the x, y, and, z data are stored one after another in a long (rows times cols time size of one element) array. Thus the three arrays have the form  $(x_{0,0}, x_{0,1}, \dots, x_{rows-1,cols-1})$ ,  $(y_{0,0}, y_{0,1}, \dots, y_{rows-1,cols-1})$ , and  $(z_{0,0}, z_{0,1}, \dots, z_{rows-1,cols-1})$ . Another example is

```
int pitch = 3*sizeof(short);
short* XYZ = (short*)(malloc(rows * cols * pitch));
SR_CoordTrfUint16(cam, &XYZ[0], &XYZ[1],
(unsigned short*)&XYZ[2]), pitch, pitch, pitch);
```

Here the output is one interleaved array of x, y, and z data:  $(x_{0,0}, y_{0,0}, z_{0,0}, \dots, z_{rows-1,cols-1})$ .

### 11.1.3 Control parameters for image capture

All API functions that can affect the cameras behavior have the form `SR_Set*()`, where `*` is a wildcard. The wrappers in the other interfaces (Matlab, Python, etc) have similar naming conventions. This subsection describes the parameters that set different control modes. There are different functions related to the control and software embedding.

With the function `SR_SetMode()` described in the previous subsection (see subsection 11.3) the flags `AM_HW_TRIGGER` and `AM_SW_TRIGGER` (these are part of the enumeration `AcquireMode`) can be used to switch between one of the triggered capture modes that are described in chapter 10.

The functions `SR_SetCallback()` and `SR_GetDefaultCallback()` are related to a callback mechanism that gives the user's process control in case of any exceptions. This means that possible occurrences of error message boxes are suppressed. So if you want to program an application which under no circumstances gets interrupted by requesting user input then you have to use callbacks to make the boxes disappear.

#### ✓ Important

*If you want to suppress any user feedback requests (e.g. due to Ethernet problems) then use the callback mechanism to suppress the message windows!*

There can be cases where the program flow is blocked. For instance, when the hardware trigger is used but the actual trigger did not appear prior to a call to `SR_Acquire()`. In order to cope with such cases the function `SR_SetTimeout()` can be used. This timeout is used at reading and writing on USB or Ethernet port. Any function which communicates with the camera will fail if it exceeds the timeout.

### 11.2 Parameters affecting measurement quality

The measures per pixel are distance (in terms of the 14 bit phase shift angle), amplitude,  $(x, y, z)$  coordinate vectors and the confidence value. The following parameters and operations modes can affect at least one of these.

### 11.2.1 Integration Time

The integration time is the length of time that the pixels are allowed to collect light (see chapter 6). Four samples are taken to produce the phase measurement, requiring four separate integration periods. Therefore the total time required to capture a depth image is four times the Integration Time, plus four times the readout time. This is reflected in the achievable frame rate for a given integration time. The integration time can be set with:

```
int SR_SetIntegrationTime(SRCAM srCam, unsigned char intTime);
```

where the second argument `intTime` is a value from 0 to 255 that sets the integration time  $IT$  in  $ms$  according to the following formula:

$$IT = 0.300 \text{ ms} + (\text{intTime}) \cdot 0.100 \text{ ms} \quad (11.1)$$

The frame rate, here denoted as  $FR$  depends on integration time

$$FR = \frac{1}{4 \cdot (IT + RO)} \quad (11.2)$$

where  $RO \approx 4.6 \text{ ms}$  is the read out time. By default the parameter `intTime` is set to 30 (0x1E in hexadecimal). This leads to an integration time of  $IT = 3.3 \text{ ms}$  and to an effective frame rate of  $FR \approx 31.5 \text{ fps}$ . The fastest frame rate is limited by the transfer of the image to the host PC. This time depends on the camera used and is around 6  $ms$ .

Too high integration times lead to saturation while to low integration times lead to noise. Higher integration times also cause higher temperature of the sensor and therefore the heat sink requirements in section 4.1 become even more important. See chapter 7 for a more detailed explanation of these interdependencies.

#### ✓ Important

*The SR4000/SR4500 has a new feature, called dual integration time that addresses the problem of saturation vs. noise.*

The dual integration time mode helps out of this problem. It effectively takes two images, one with a higher integration time and one with a lower integration time. Saturated pixel values in the higher integration time image are replaced by (hopefully not saturated pixels) of the lower integration time image. It follows that the frame rate is higher if the feature is switched off. The dual integration time feature can be set with:

```
SR_SetDualIntegrationTime(cam, ratio);
```

where the variable `ratio` defines the ratio of the upper and lower integration time. The upper integration time is set by `SR_SetIntegrationTime()`. `Ratio` is a percentage value in a range of 0 to 100. If the ratio is zero the camera operates with single integration time. The single integration time mode is set as default.

The function `SR_GetIntegrationTime()` retrieves the current integration time.

### 11.2.2 Amplitude threshold

The parameter `amplitude` defines a minimum amplitude that needs to be exceeded in order to accept a measurement. This parameter can be used to suppress values with low amplitude that might be unstable. The amplitude threshold can be set with

```
int SR_SetAmplitudeThreshold(SRCAM srCam, unsigned short val)
```

where the value `val` ranges from 0 to  $2^{16} - 1$ . The effect is that the distances and the  $(x, y, z)$  values of a pixel with a lower amplitude than `val` are all set to zero.

For static scenes, the reflected signal amplitude can be used as a measure of the quality of corresponding distance measurements alternatively to the confidence value. In contrast to the confidence map the amplitude image needs no extra processing resources.

The function `SR_GetAmplitudeThreshold()` retrieves the current amplitude threshold. The default value is zero.

**✓ Important**

*Note, that for the NARE feature of the SR4500 to work the amplitude threshold must be set to an appropriate value. Such a value is set by default when using the SR4500.*

### 11.2.3 Modulation frequency

The parameter modulation frequency can be set with the function

```
int SR_SetModulationFrequency(SRCAM srCam, enum ModulationFrq modFrq)
```

where the enumeration type has the following assignment as shown in table 11.1.

modFrq	1	6	8	9	10	11
MHz	30	15	29	31	14.5	15.5

Table 11.1: Type assignment for the modulation frequency selection.

The modulation frequency can be changed by the user mainly to enable for multi camera operation (see section 8.4 on page 26). This may be important in applications where more than one SR4000 camera is used e.g. to capture a larger scene with the sensors.

**✓ Important**

*Note that the change of frequency for multiple camera operation is not necessary when using the SR4500. The SR4500 has a special built-in feature that allows for multiple camera settings.*

These frequency settings are chosen such that the neighboring frequencies are different enough to avoid mutual interferences but close enough that the camera calibration is still valid. However, generally the following should be noted:

**✓ Important**

*If a camera is used outside it's specified frequency then the specifications given in the product data sheet do not apply anymore as the measures are affected by the modulation frequency.*

### 11.2.4 Auto exposure

The SR4000/SR4500 API provides a controller that adapts the IT on-line by ensuring that no more than a preset maximum fraction of pixels is saturated. This leads to the highest frame rate possible given the saturation constraint. The function `SR_SetAutoExposure()` enables for this mode. Please



refer to the HTML API description for further information on the usage of this function.

### 11.3 Filters and auxiliary functions

#### Filter modes

Some filters and auxiliary functions are provided:

- **Median filter:** (AM\_MEDIAN). This turns on a 3x3 median filter (see e.g. [http://en.wikipedia.org/wiki/median\\_filter](http://en.wikipedia.org/wiki/median_filter) to understand a median filter. Note, this algorithm is run on the host PC and not on the camera. By default this mode is disabled.
- **Convert gray mode:** (AM\_CONV\_GRAY). This mode compensates for the light intensity send out by the sensor itself. The resulting image is more looking like a conventional (passive) gray-level camera image. This mode is processed on the host PC and requires for some processing power. The mode is set to active by default.
- **Confidence map:** (AM\_CONF\_MAP). This flag has to be enabled to compute the Confidence Map (see section 9.4 on page 9.4 for a description of this map). Only when this mode is set then the confidence image is available with a call to SR\_GetImage(). By default this mode is disabled.
- **Adaptive Neighborhood Filter** (AM\_DENOISE\_ANF). This value turns on a hardware-implemented noise filter. This filter combines amplitude and distance information in a 5x5 neighborhood. It reduces noise while preserving detail such as edges or small structures, with no computational cost to the host CPU. The filter is active by default.

All these modes are set with the function

```
int SR_SetMode(SRCAM srCam, int mode)
```

that takes the argument mode of type int. This is a combination of the flags specified in the enumeration AcquireMode (see the HTML API documentation of the SR4000). Many mode flags can be combined with a bitwise OR operation. Each possible setting has the form AM\_\*, where the \* is a placeholder. Note that the function overwrites the modes set previously. So to set the mode safely a function like the C-function srSetMode() shown here:

```
int srSetMode(int currentMode, int flag, bool onOff)
{
    int mode = currentMode & (~flag);
    if(onOff) mode |= flag;
    return mode;
}
```

must be used in order to not overwrite previously set values or default values.

For instance, to set the median filter on one can write:

```
int SR_SetMode(cam, srSetMode(SR_GetMode(cam),AM_median,true)
```

and to disable it:

```
int SR_SetMode(cam, srSetMode(SR_GetMode(cam),AM_median,false)
```

12 Driver installation and setup

It is recommended to install the software first, and then to plug in the camera. The installation procedure is different depending on Operating System, and version. The different procedures are outlined below. The following figure (figure 12.1) shows the installation sequence for Windows Vista (32-bit).

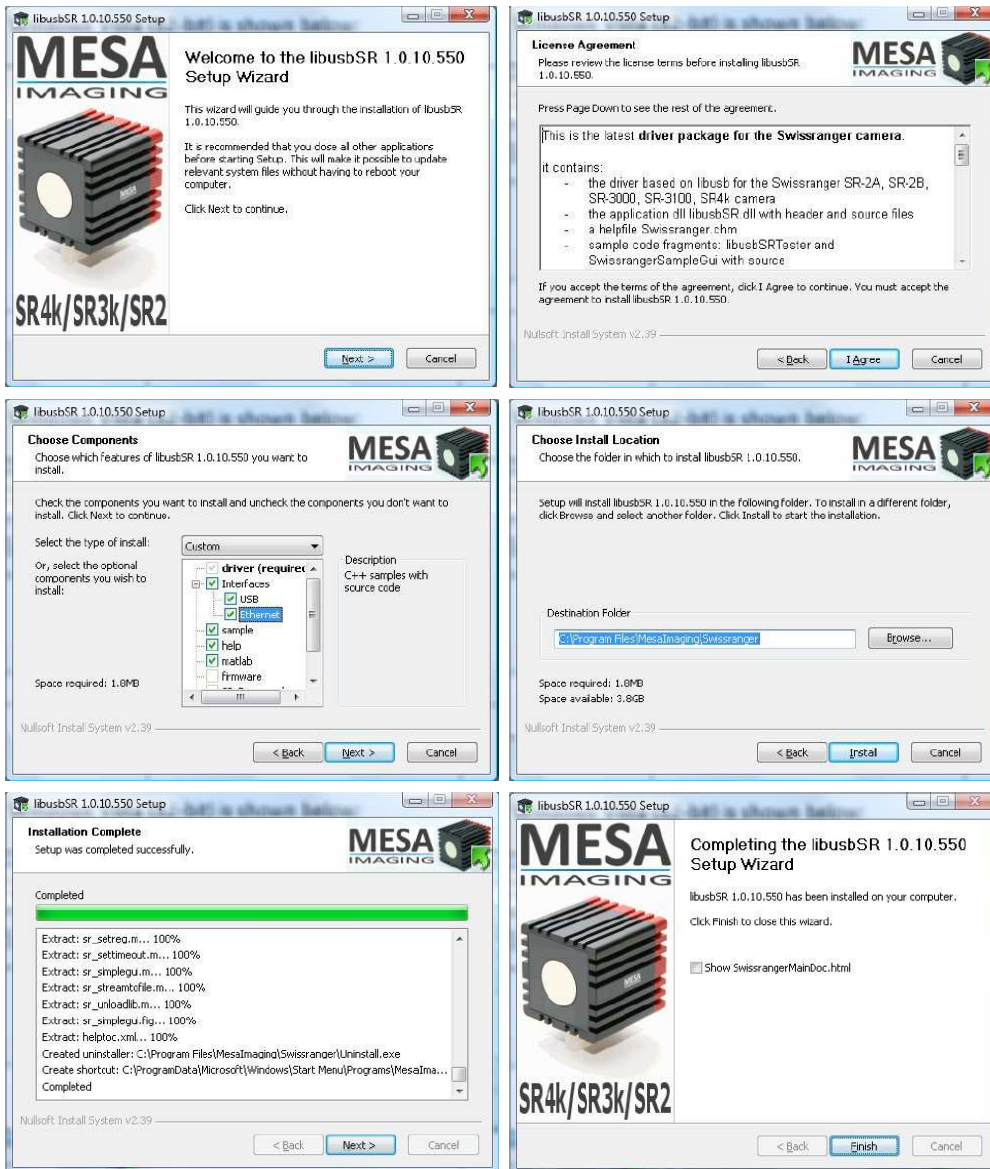


Figure 12.1: The sequence of steps (from left to right in each row, from top to bottom over the rows) that installs the software on windows.

## 12.1 Installing driver, demo and sample software

Supported operating systems are the Windows 32-bit platforms (XP/Vista/7). A beta version of a Linux driver is also available, but driver updates will be done on a less regular basis compared to the Windows driver.

### 12.1.1 Windows

The Setup program may run automatically when the CD is inserted. If not, run the Installer, e.g. *SwissrangerSetup1.0.10.550.exe* from the root menu of the CD. Alternatively, run an updated version of the Setup file downloaded from the Mesa website:

<http://www.mesa-imaging.ch/drivers.php>

On Windows Vista/7, some security confirmations must be given by the user during the installation process. After successful start of the installer the sequence shown in figure 12.1 (previous page) can be followed.

To enable support for Ethernet cameras, the Interfaces node must be expanded, and the Ethernet check box checked.

### 12.1.2 Linux

The installation packages may be found on the CD under Linux. Alternatively, please go to the Mesa Imaging Website for the latest installation packages and instructions on how to install the SwissRanger software on your system:

<http://www.mesa-imaging.ch/drivers.php>

## 12.2 Uninstalling and reinstalling the driver software

Sometimes issues with the driver software can be solved by reinstalling the driver, or installing an updated driver version. Before doing either, it is important to unplug the camera and uninstall the driver software. This is done as follows:

1. Select Start→Programs→MesaImaging→SwissRanger→Uninstall and follow the uninstall wizard (see figure 12.2, below).
2. The driver may then be reinstalled as described above.



Figure 12.2: Uninstall wizard

## 12.3 Connecting a USB camera

The power and USB may be connected in either order. Once power is connected, the green Indicator LED will start flashing slowly which indicates that the camera has successfully initialized.

### 12.3.1 Windows

A tool-bar notification (see figure 12.3, below) is displayed indicating that the device driver software is installing, and then another indicates that the device driver installation has completed successfully.



Figure 12.3: This note appears after successful installation in Windows

If the device driver has installed successfully, any connected cameras will appear in the device manager under the category LibUSB Devices. Note that this applies to USB cameras only.

## 12.4 Connecting an Ethernet camera: using DHCP or static IP address

The order in which power and network cables are connected is important for Ethernet SR4000, since this determines whether a fall back IP address is used, or the normal DHCP or static address.

### 12.4.1 Connecting to an SR4000 using DHCP

SR4000 Ethernet cameras are shipped with DHCP enabled by default. Simply connect the camera to a network which has a DHCP server and then apply power to the camera. The camera obtains its address via DHCP.

### 12.4.2 Setting a static IP address

Instead of using DHCP, it is possible to set a static IP address. In order to do so, it is necessary to set the static IP address of the host PC first.

In order to set the static IP address of the camera, it is necessary to use a fixed fall back address:

- Apply power to the unit without connecting the Ethernet cable.
- Wait 20 seconds before plugging in the network cable.

The camera now has a fixed IP address of 192.168.1.42. The camera cannot function normally using this address: the only purpose is to set another static IP address.

In the following, the tool Telnet will be used. This is not enabled by default on Windows platforms. Here are the instructions to do this:

1. Start→Control Panel→Programs and Features
2. On left side, select "Turn Windows features on or off"
3. Check Telnet, wait until it has been installed

Start now a command prompt or equivalent console. Windows Vista users, see note on Telnet below.

1. Telnet to the camera: `telnet 192.168.1.42`

2. set the static address, e.g. `fw_setenv staticip 192.168.1.33`

Exit telnet and re-power the camera with the network cable connected. The camera will always use the static address on power-on when attached to a host PC or network.

### 12.4.3 Reverting to DHCP

To set the camera to use DHCP again, the static IP address must simply be cleared as follows:

1. Telnet to the camera (e.g. using the fall-back address mode as described above): `telnet 192.168.1.42`
2. clear the static address: `fw_setenv staticip` (no argument clears any existent IP)

Exit telnet and reboot the camera with the network cable connected. The camera will always use DHCP to get an address on power-on, when attached to a network with a DHCP server.

When using DHCP or static IP, the addresses of the connected cameras are visible in the Camera Selection Dialog Box of the software applications, as described in the next section.

Windows Vista: Setting a static IP address on the host PC

1. Start→Control Panel→Network and Sharing Center
2. Right Click on your Local PC Name (This Computer)→Properties
3. Select Internet Protocol Version 4 (TCP/IPv4), Click Properties
4. Select radio button Use the following IP address
5. Set IP Address, (e.g. 192.168.1.xx where xx is any other than 42 to communicate with a camera using the fall back address.)
6. Click on Subnet mask, to apply default of 255.255.255.0
7. Click OK
8. Click Close

### 12.5 Silent install in Windows

Sometimes, one wants to install the driver in Windows “silently”, i.e. without the user feedback required. A so-called *silent install* can be done by the command:

```
SwissrangerSetup1.0.14.747.exe /S
```

The number 747 might be different when the latest driver is used, the /S indicates the silence. However, newer Windows versions require for the confirmation that you trust MESA Imaging as software provider. Then the installer needs to be invoked as administrator and a further OK mouse click is required. In this case automated silent install can be done with a script. Please contact our customer support (<mailto:support@mesa-imaging.ch>) to receive an example of a VBS script (Windows).

#### ✓ Important

*For the SR4500 to work properly a driver version 747 or later is required!*

### 13 Software applications

Once the driver software has been installed and the camera is connected, a software application may be used to access the camera. In this Quick-start section the use of the applications SwissRangerSampleGUI and SR\_3D\_View will be explained.

#### 13.1 Using the SwissRangerSampleGUI Application

This section applies to the SwissRangerSampleGui application which is available only on Windows. However, the information is relevant to Linux users also, since it describes the types of image produced by the camera and the most important adjustments available to the user.

SwissRangerSampleGui can be launched from

Start→Programs→MesaImaging→Swissranger→Sample SwissRangerSampleGui

*A notification may appear at this stage advising of a new driver dll version available on the MESA website. This may be downloaded by clicking on the link that appears.*

The SwissRangerSampleGUI Interface is shown in figure 13.1.

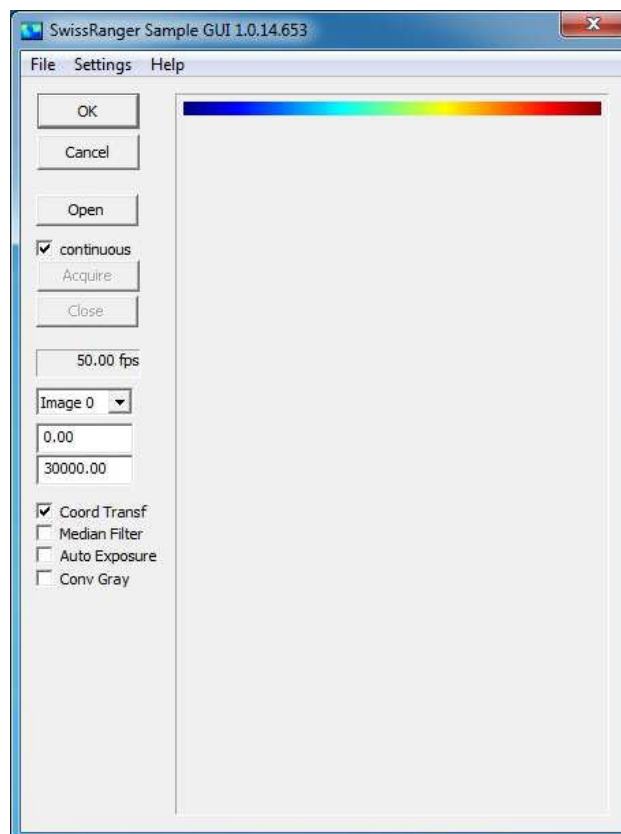


Figure 13.1: SwissRangerSampleGui interface prior to opening a camera.

To connect to a SwissRanger camera, click the Open button: **Open**.

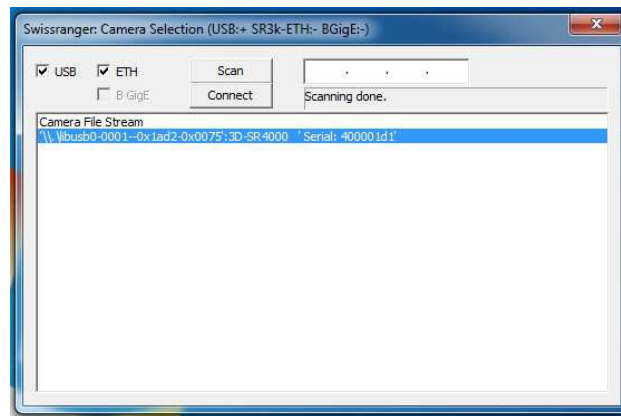


Figure 13.2: Open dialog: select a camera (or a file stream) with a mouse click. The camera text must be blue backgrounded in order to proceed.

A dialog will appear showing the available SwissRanger cameras (see figure 13.2, above). In this example there is one USB camera listed and selected. An already-connected Ethernet camera is marked with a \*. Note that “Camera File Stream” is used to replay camera data files previously recorded using another application.

To proceed, select a camera from this list and click

**Connect**,

or simply double click.

**✓ Important**

*If the camera has already been connected, the camera selection dialog box will not appear.*

The application will automatically connect to the same camera as the previous time, if this camera is available. In order to force the dialog box to open, one must select the Open (force dialog) in the File Menu of the GUI.

To start acquisition, click the acquire button:

**Acquire**.

The camera will now start continuous acquisition, with the acquired images being displayed in the application window. The images are displayed in false color, with a color bar above them which indicates the color scale from blue (zero or minimum) to red (full-scale or maximum).

Below the buttons **Acquire** and **Close** the frame rate is displayed. In the next section the different types of image data produced by the SR4000 as well as the most important adjustments are explained.

**13.1.1 Distance image**

The distance image contains distance values for each pixel. Depending on whether the **Coord Transf** check box is checked, this is either radial distance from the camera (set by default), or the z values as described in chapter 9. In the SampleGUI application the distance values are scaled along the color spectrum that is shown in the top of the display area, right to the **OK** button.



### 13.1.2 Grayscale image

The illumination striking an object decreases with the square of the distance from the camera to the object. Therefore, the signal amplitude is much lower for more distant objects. The check box **Convert Gray** activates a mode that compensates for this effect to produce a gray scale image similar to that of conventional gray-level cameras. When this mode is off, this image is simply the amplitude signal (see chapter 9).

### 13.1.3 Scaling of display color

Below the frame rate textbox is a drop-down control which allows selection of image0 and image1, corresponding to the distance or gray scale image. The two edit boxes below this drop-down can be used to set the display color range. The top edit box is the minimum range value. Any value at or below the value entered is displayed dark blue. The lower edit box is the maximum range value. Any value at or above the value entered is displayed dark red. All values in between are displayed with the proportional value on the color scale.

### 13.1.4 Important adjustments

From the Settings menu, select Settings (dll dialog). This causes a dialog to be displayed which contains two sliders. These can be used to control integration time and the confidence threshold.

#### Integration time

This is the most important camera control available in the demo application. The integration time can be adjusted using a slider control in the Settings→Settings (dll dialog)... dialog box. Adjusting this value controls how long each sensor pixel collects light (see chapters 6 and 7).

#### Amplitude threshold

Amplitude by itself can be used as a measure of the quality of corresponding distance measurements. It can be adjusted using a slider control in the Settings→Settings (dll dialog)... dialog box.

### 13.1.5 Acquisition options

Below the edit boxes used for the image color scaling are a set of four tick boxes:

- **Coord Transf**: switches between radial distance and distance along the optical axis (z-value).
- **Median**: applies a 3 by 3 median filter to the distance data
- **Auto Exposure**: adjusts the integration time depending on the maximum amplitudes present in the image.
- **Conv Gray**: produces a distance-adjusted gray-scale image.

## 13.2 Visualizing and capturing 3D data using the SR\_3D\_View Application

The SR\_3D\_View application is installed by default when installing the driver from the CD. Alternatively it may be downloaded from the MESA Imaging website at

<http://www.mesa-imaging.ch/demosoftware.php>

This software depends on DirectX being installed on the host PC (version 9c minimum).

To connect to a SwissRanger camera, click the Start button: **Start**. In order to enter the opening to select a camera manually dialog hold the **Ctrl** key and press the button **Start**. The button **Start** will change to **Stop** (as can be seen in figure ??, ❶).

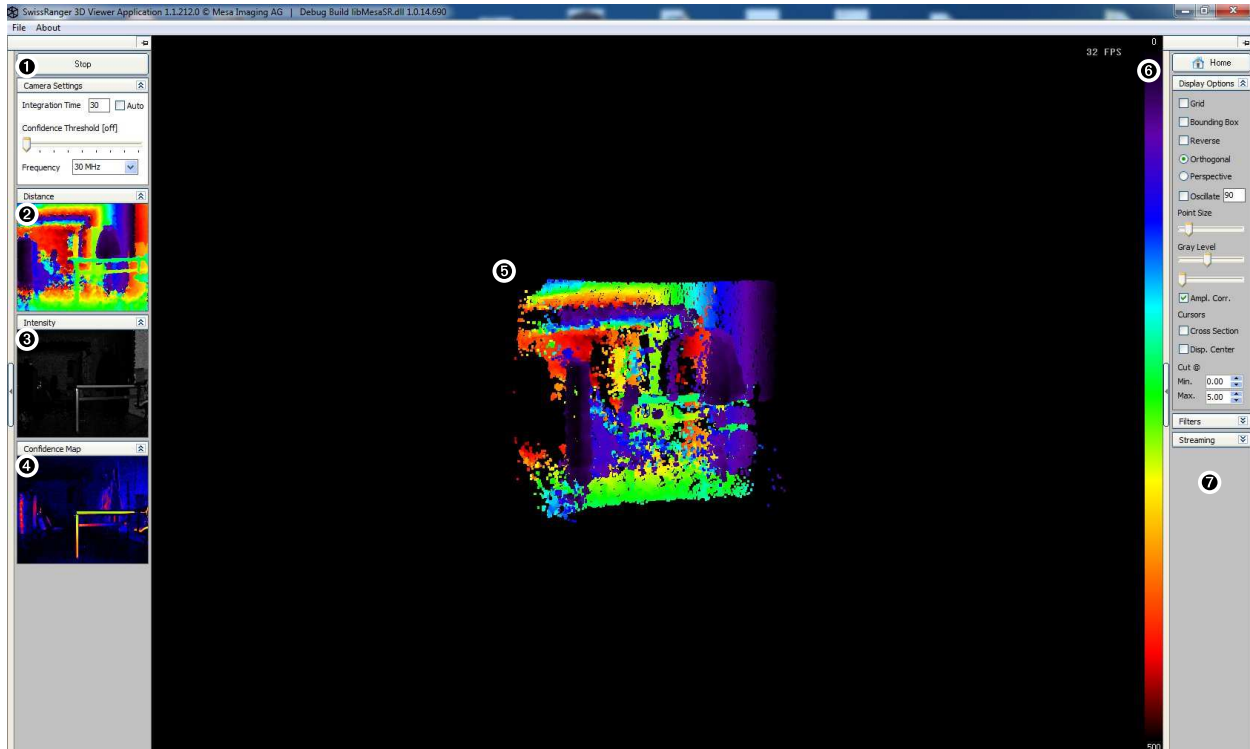


Figure 13.3: The SR\_3D\_View Interface showing distance, gray scale and confidence map images on the left hand side and 3D projection in the main window.

There are a few control items below ❶. The integration time can be adjusted with an edit window below **Camera Settings** to avoid saturation and to achieve the desired balance between low noise and high frame rate. The check button **Auto** activates the auto exposure function (see also chapter 11) that adapts the integration time on-line. Further controls on the left side of the window include a confidence threshold and frequency selection. The confidence threshold can be changed to suppress lower confident pixel values. The frequency can be changed to change the full phase distance. However, note that the camera is not calibrated for frequencies other than its nominal frequency.

As shown in figure 13.3, the distance (❷), gray scale (❸) and confidence map (❹) images will appear on the left hand side of the main window. Those images are unadjusted raw pixel arrays, whereas the 3D projection of the central image (❺) has been adjusted by the coordinate transformation function in the driver to compensate for radial distortion in the optics, removing the “pin cushion” effect. Here the full  $(x, y, z)$  coordinate vectors are displayed. The coloring of the points can be selected as a color proportional to the  $z$ -value of a pixel (see the color bar ❻ that shows the color spectrum from zero to full phase distance) or as amplitude values.

The 3D representation in the large center screen (❺) may be oriented by holding the left mouse button while moving horizontally or vertically or by using the cursor control keys. Zoom is controlled by the mouse scroll wheel or Page Up/Page Down keys.

For a summary of the main controls press the **F1** key.

Elements of the main window may be configured from a menu which is activated by a right mouse button click. These are:

- **Solid**, **Wire Frame**, and **Point List**: different display 3D point display options.
- **Color** or **Amplitude**: selection of the 3D overlay source.
- **Cross Section**: draws a line through a single row and column of the data. The position of the line corresponding to Point A may be selected by clicking on the distance (②) or grays scale image (③) with a left mouse button click. The position of point B may be selected on the same images by using the right button. The coordinates of the points and the distance are shown on the top/left of the 3D area.
- **Background** and **Load Image...**: background selection of the 3D area (④).

On the right hand side of the screen there is a column (marked with ⑦) with a button **Home** that resets the perspective changes that were done in the 3D area. Furthermore, there are the following three pull down menus: **Display Options**, **Filters**, and **Streaming**. Only one of these can be opened at a time. The three pull down menus and further control keys are described in the following subsections.

### 13.2.1 Display options

In the display options the check buttons **Grid** and **Bounding Box** draw a ground plane and an indication of a smallest enclosing box with fix orientation. The check button **Reverse** flips the image in terms of the x-values (mirror effect).

The radio buttons **Orthogonal** and **Perspective** switch between the 3D mode (perspective) and a mode in which the z-values are shown as an orthographic projection of the pixel array (orthogonal). In the orthogonal model the z-values can be stretched and shifted by holding the **Shift** key (stretching) or the **Ctrl** key (shifting) together with the left mouse and horizontal mouse movements.

The check button **Oscillate** makes the scene moving automatically and periodically left and right for demo purposes.

The slider **Point Size** adjusts the size of the 3D points drawn in the 3D area (⑤).

Two sliders below **Gray Level** control an amplitude offset and scaling that affects the grays scale image (③) on the left of the screen. The check button **Ampl. Corr.** automates this adjustment.

There are two 3D cursor control items. The **Cross Section** enables for the same functionality as is available in the right mouse click context menu. The **Disp. Center** offers to change the 3D center of rotation.

Color scaling for color coded distance is selected using left mouse button and a drag and drop mechanism applied to the top and the bottom of the vertical color bar (⑥) right to the 3D display.

**HOME** on the keyboard brings the 3D scene back to the default view.

### 13.2.2 Filters

The filter options allow for three kind of filters. First, the **Median** check button applies a median filter with a 5x5 neighborhood. Second, the **ANF** activates an *adaptive neighborhood filter*. This filter takes into account a stability measure computed on the local neighborhood of a pixel. The median and the ANF filter are both enabled by default. The third filter, activated with the **FIFO** check button, is a

simple time averaging filter that can be used to improve the measurements in static scenes. An editing window right to the **FIFO** button allows to set the number of past frames that get averaged.

### 13.2.3 Streaming

The streaming section of the interface enables the capture of a number of frames of data. There are three formats supported:

- ASCII: output of the frames in separated text files (see box “ASCII format” below).
- AVI Video: an AVI video that can be watched with conventional video playing software (e.g. VLC or Windows Media Player).
- SRS: SwissRanger stream as described in chapter 11. This can be opened as a camera in the opening dialog.

**ASCII format:** *The SR\_3D\_View application can export a data stream from a SwissRanger camera containing 3D point coordinates and amplitude data.*

*Each captured range image in the sequence is saved in a separate file, with the filenames being numbered name\_0001.dat, name\_0002.dat etc.*

*Within each file, z, x and y coordinates are arranged as arrays of 144 rows of 176 tab delimited floats. Coordinates are in meters. The coordinate system is Right Handed, with z being the distance along the optical axis from the front face of the camera, and from the camera's viewpoint, x increasing to the left and y increasing upwards, with zero x and y lying on the optical axis.*

*This is followed by 144 rows of 176 tab delimited integers for the amplitude image. Amplitude full-scale is 0x7FFF.*

*Each array is preceded by a description e.g.*

*Calibrated Distance' indicates the Z coordinate array.*

*Calibrated xVector' indicates the X coordinate array.*

*Calibrated yVector' indicates the Y coordinate array.*

*Amplitude' indicates the amplitude array.*

*Confidence map' indicates the Confidence Map array.*

*If a threshold on amplitude has been applied, all 3D coordinates and amplitudes will be zero where the amplitude is below the threshold for the corresponding pixel.*

All formats are recorded with setting a file name by pressing **Set Filename** followed by **Start Streaming** and **Stop Streaming**. If an existing file is about to be overwritten there is no warning. An SRS stream can be re-played by opening the camera file stream option in the opening dialog (hold **Ctrl** and click on **Start** to enter this dialog).

### 13.2.4 Further controls

#### Capturing single frames

The menu option **File** above **1** offers a few export functions for single frames. A frame may be captured and exported in STL, DXF or point cloud format in the following way:

When the camera is streaming live data, the **Start** button at the top left becomes a **Stop** button. Clicking on this will freeze the data for the most recent frame. This may then be exported by clicking on the **File** menu, and selecting the desired format.

### Control keys

**Ctrl + F12** shows the camera parameters dialog box which also allows adjustment of integration time and amplitude threshold.

**Ctrl + R** performs a left-right flip of the displayed images.

### 13.3 Sample Code

Example code is provided in C++ with a simple GUI for Windows in the `SwissrangerSampleGUI` sample, and without GUI for Windows and Linux in the `LibusbSRTester.cpp` 'console' application.

Some changes may be needed for the code to compile on different development platforms. Project files are included for Microsoft Visual Studio 2005, and Makefiles are included for Linux.

### 13.4 Matlab Interface

A Matlab interface to the SwissRanger is provided for Windows. This is installed by default during driver installation to

```
C:\Program Files\MesaImaging\Swissranger\matlab\swissranger
```

To make the SwissRanger interface available within Matlab, this path must be added to the end of the Matlab search path using the Matlab Set Path command. The section of the Help file 'Matlab Interface to SwissRanger Cameras' contains more information. For an overview of the SwissRanger m-functions use the command `help SwissRanger` within Matlab.

### 13.5 C# Interface

A code example for a C# interface is available from [www.mesa-imaging.ch/temp/SwissRanger.zip](http://www.mesa-imaging.ch/temp/SwissRanger.zip).

## 14 Help and Support

For aspects of the setup and operation of the camera, refer also to the Knowledge Base and FAQ on the MESA website [www.mesa-imaging.ch](http://www.mesa-imaging.ch) or refer to our forum <http://forum.mesa-imaging.ch>. Alternatively, you can send a question to the MESA support team: <mailto:support@mesa-imaging.ch>.

## 15 Electrical power and trigger connection

### ✘ WARNING

*Only use the camera together with the power supply that is supplied by MESA imaging.*

In case that another power supply is used, observe following rules:

- Power supply isolation from the ground voltage must be redundant (i.e. double isolated power supply).
- Select and implement wire cross-sections and their correct fuse protection as per the applicable standards.
- Design a power supply that is conform to chapter 2.5 of EN 60950-1:2006 (risk of fire due to electrical dysfunction).

### 15.0.1 SR4000

All SR4000 cameras include two Lumberg M8 connectors. One connects the power supply and another connects to trigger in- and output signals. The pin assignments are shown in figure 15.1.

The communication cable connectors (USB or Ethernet) are standard ones.

The camera also includes a status LED. Regular pulsing of the status LED indicates that the camera is powered; fast pulsing of the status LED indicates data transfer between camera and computer.

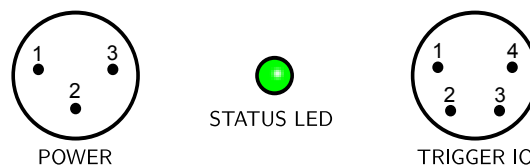


Figure 15.1: Schematic view of the connectors on the backplane of the camera including numbering of pins

Table 15.1: - Power Connections SR4000 -

1	+12 VDC; Min -2%; Max +10%	Typ. 0.8 A @ 12 V, Min 0.6 A , Max 1.0 A
2	SHIELD	Connect to protective earth
3	GND	Ground

For more details on the trigger logic, please refer to chapter 10, section 10.2.1 on page 32

### 15.0.2 SR4500

The SR4500 is equipped with an industrial M12 power socket (IP67). The pin assignments are shown in figure 15.3.

Table 15.2: - **Trigger I/O Connections SR4000** -

1	External Voltage	4.5 V - 5.5 V / 10 mA	defines the logic level of the trigger output
2	Trigger In	4.5 V - 5.5 V / 15 mA	Start acquisition (external voltage level)
3	Trigger Out	4.5 V - 5.5 V	Frame ready to fetch
4	External GND		In reference to External Voltage

### TRIGGER SIGNALS

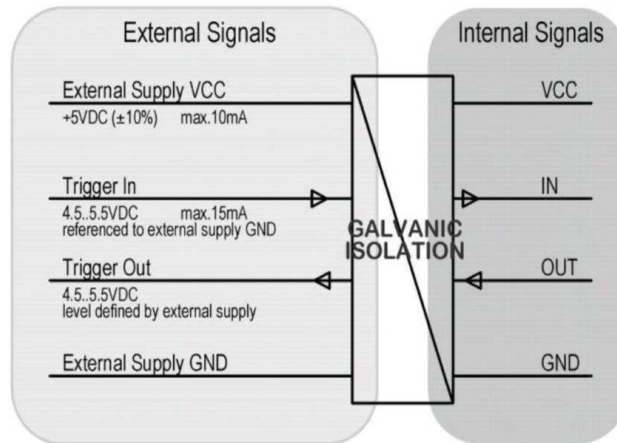


Figure 15.2: Schematic view of the hardware trigger logic

The communication socket is a standard industrial Ethernet connector.

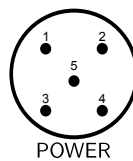


Figure 15.3: Schematic of the SR4500 power connector



Table 15.3: - **Power Connections SR4500** -

1/3	+24 VDC; Min -10%; Max +10%	Typ. 2.0 A @ 24 V, Mean 0.5 A , Max 5.0 A
2/4	GND	Ground
5	NC	Not connected

## **16 Ratings and conformance**

Please refer to the official data sheet or specification sheet of your product.