FAQ
3D Cameras

Document History

Version	Comment	Date
1.00	Created	2008-09-16
1.01	Minor changes	2008-10-31
1.02	Minor changes	2008-11-17
1.03	Added FAQ O21	2009-02-12
1.04	Updated FAQ C08	2009-02-26
1.05	Added FAQ C16	2009-05-06
1.06	Updated C16	2009-05-13
1.07	Added FAQ U01, O06, O09, O20, H06	2009-09-04
1.08	Added FAQ C06, I20, L02, O08, O10, O11, O16, P06, T04, T05, T06, U04	2010-02-24
1.09	Updated FAQ C02	2010-03-02
1.10	Added FAQ H09	2010-06-08
1.11	Added FAQ H07	2010-06-21
1.12	Updated FAQ H06	2010-08-31
1.13	Added FAQ H03	2011-08-17

Table of Contents

Connectivity

C02 – How should I connect my Ranger D/E or Ruler E to a PC?

Hardware

The connection is required to be via a gigabit Ethernet network. Although the throughput may be less than 100 Mbit/s in many configurations, the system requires gigabit Ethernet. Connections using 10 or 100 Mbit/s networks **will cause problems and is definitely not recommended!**

The cameras should be connected via a **dedicated network.** Do not connect other networking devices to the same network. The traffic from the camera is unacknowledged UDP packets since there is no time for resending of data in high speed applications. Notice that there is some redundancy in the data sent from the camera which means that some lost packets can be recovered on the PC side (more on this later).

Switches

You may connect via a gigabit Ethernet switch if this is beneficial to you. If you want to use the High Performance drivers for optimal performance your switch needs to support Jumbo frames (Ethernet packets larger than 1500 bytes). Unfortunately, jumbo frames handling is not formally standardized which means that there is no guarantee that all network devices obey the same specifications. There is a short list of supported switches in the manual (Appendix D of the Reference Manual). These switches have been properly tested at SICK IVP but are in no way the only switches you can use.

Connecting Several Cameras to One PC

You can connect several cameras to one PC in one of two ways. You can either connect all the cameras to one network port using a switch or each camera to its own network ports using several network adaptors or a multi port adaptor (such as the Intel PRO 1000 PT quad adaptor). At least theoretically the latter should be better from a networking point of view but there is not much evidence to support this theory. **If you connect all cameras through the same network port, the IP addresses of that port and all the cameras need to be part of the same subnet (see below). If you connect each camera to its own network adaptor, each camera/PC port pair needs to be part of the same subnet but must not be part of the same subnet as any of the other pairs.**

Software Configuration

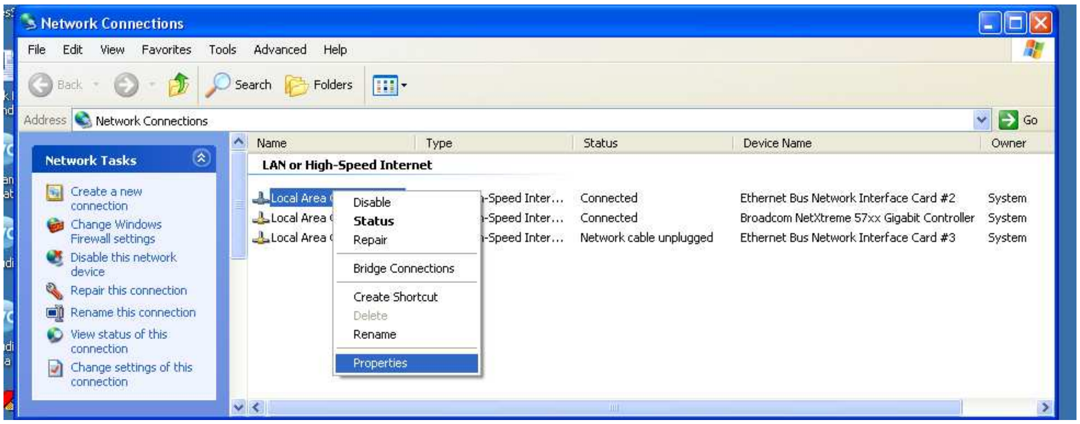
The camera and PC need to be configured to communicate on the same IP subnet in order for the communication to be routed properly from the camera to the PC. In practise this means that the camera and PC IP addresses need to be configured to talk directly on the same local network. There are several ways of configuring this but we recommend you to use the non-routable C network 192.168.x.y. Follow the step-by-step instructions below to set up a working configuration.

Firewalls

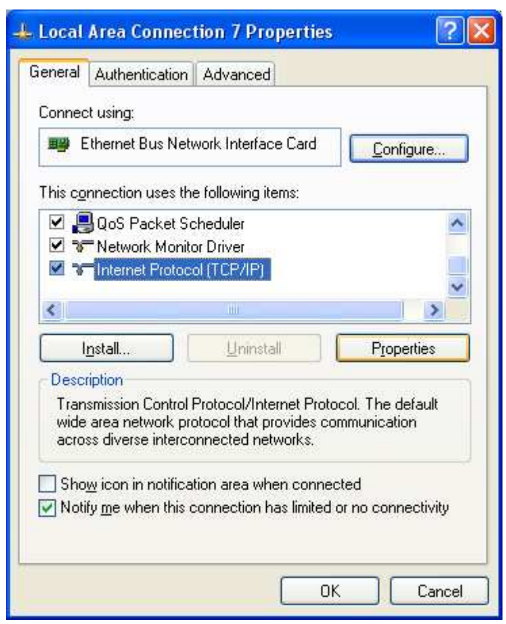
Ensure either that the network used by the camera is not controlled by a firewall or add all Ranger applications to the list of firewall exceptions. By default Windows blocks UDP traffic from unknown applications so you need to manually disable the firewall or add an exception.

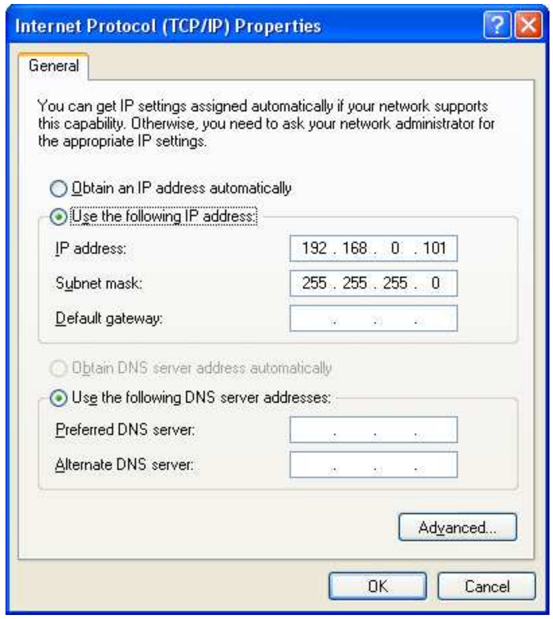
Step-by-step instruction

1. Connect the camera and PC via a switch or directly.
2. Identify which network device is associated with the camera on the PC side and edit its Properties. You will find the window below through the Control Panel.

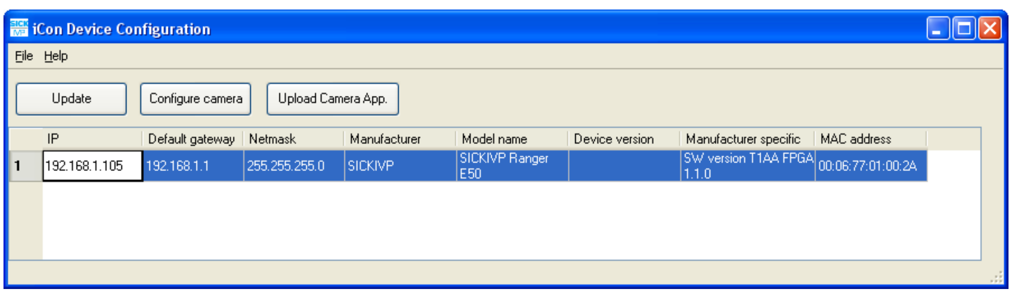


3. Select the TCP/IP item in the dialog and click Properties.

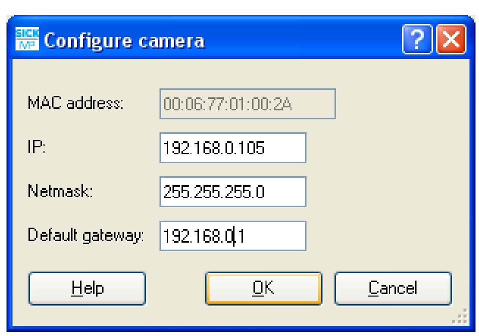




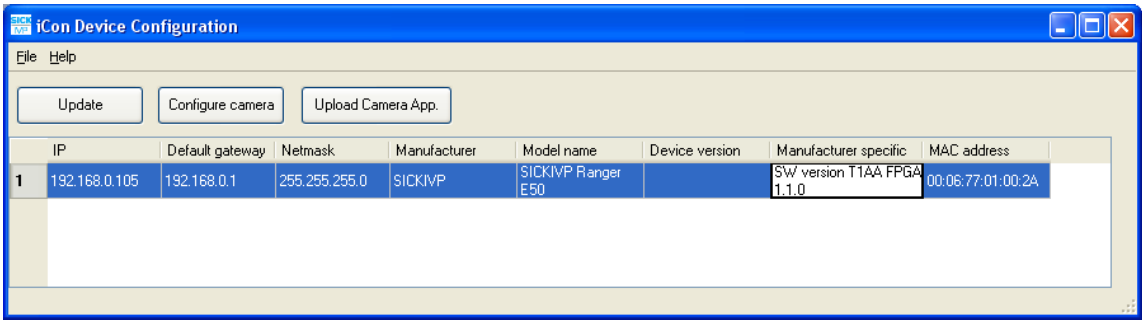
4. Set the IP address to 192.168.0.x, where x is any number between 0 and 254.
5. Set the subnet mask to 255.255.255.0. These IP and mask settings will result in a local network. IP addresses starting with 192.168.0 are never routed outside the local network and can never be reached by hosts outside the local network.
6. Start the ICON configuration tool. Make sure the camera is powered on. Click Update.



7. You should now see your camera(s) in a view like the one above. Select the camera you wish to use and click ‘Configure camera’.
8. Set the camera IP to 192.168.0.y, where y is not the same number as x. Set the subnet mask to 255.255.255.0. The default gateway is not really used since no networking should be routed anywhere outside the local network. Set it to 192.168.0.1 just to set it to something.

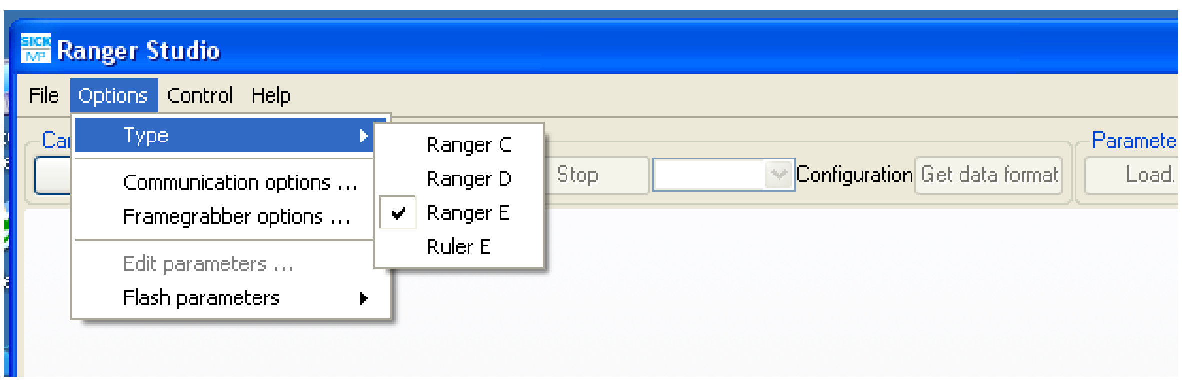


9. Click Ok and then Update in the main window of the configuration tool. You should now (after a few seconds waiting) see the camera updated with the new IP address

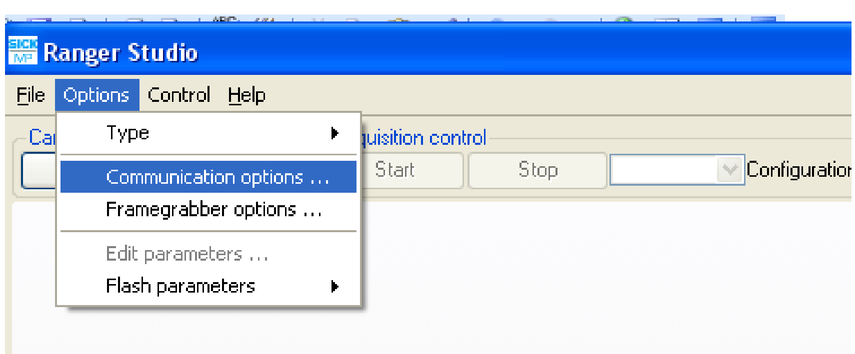


10. Make sure the firewall is disabled for the network port used by the camera or add all Ranger application programs to the list of exceptions. The Ranger needs to be able to send its UDP packets through to your Ranger application (and to Ranger Studio). If this causes any violations of your company's security policy, discuss it with your system administrator and remember that the Ranger cameras should be connected to dedicated networks where there should be no other traffic. The security threat is therefore to be treated as no more serious than connecting a USB device to your computer.

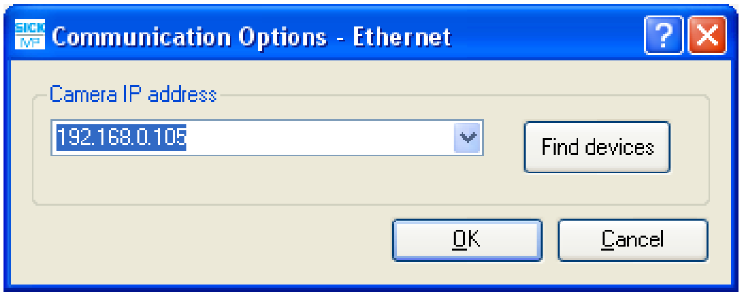
11. Start Ranger Studio. Choose the appropriate camera type.



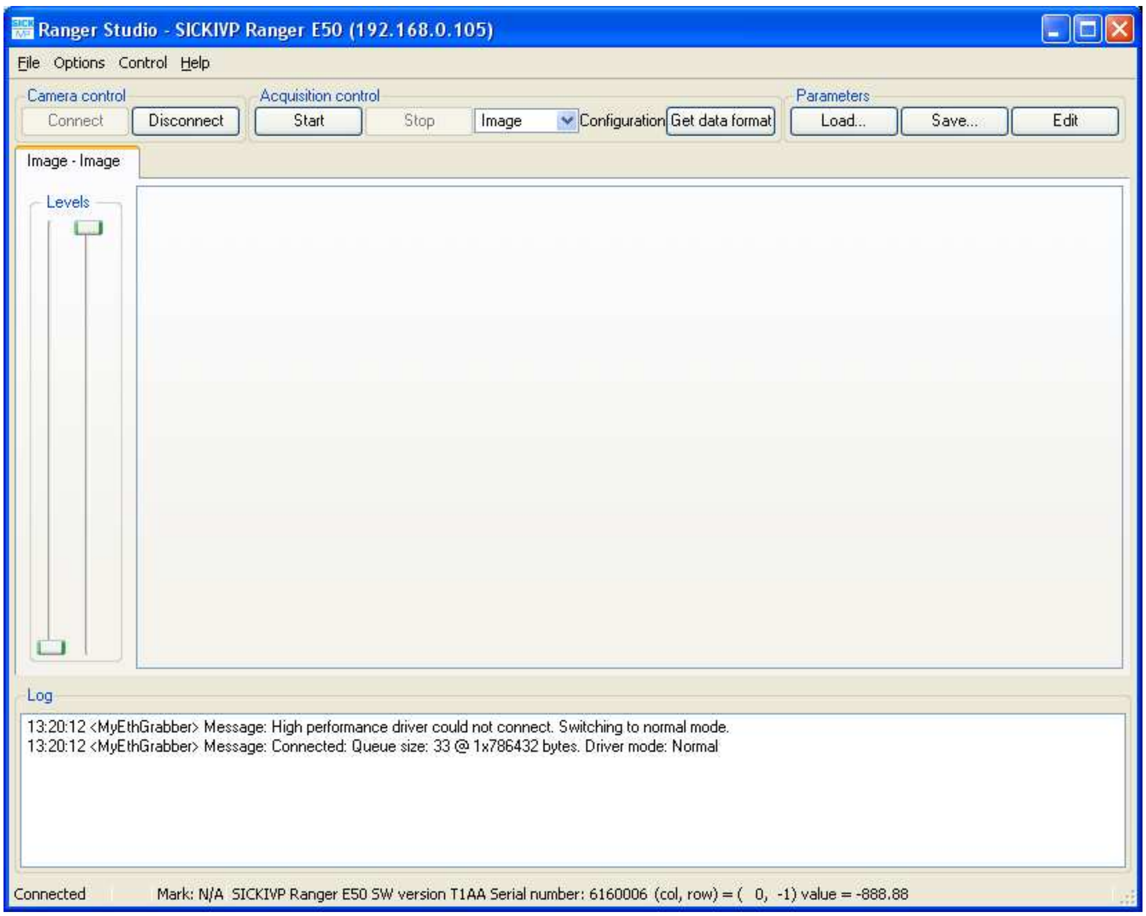
12. Enter the Communication options dialog



13. Press Find devices and look for your camera in the list of IP addresses. Select it and click OK



14. Click Connect in the main window. You should now see a view like this:



If you have enabled the High Performance drivers the message in the log window will be different. If the high performance drivers were successfully loaded the last row will end with the message Driver mode: fast.

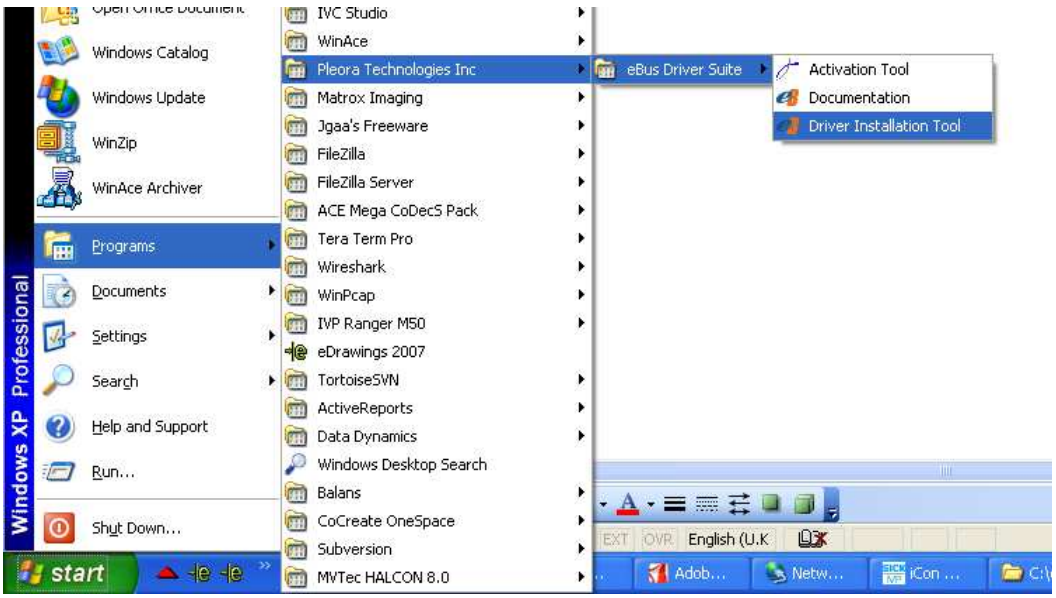
High Performance Drivers

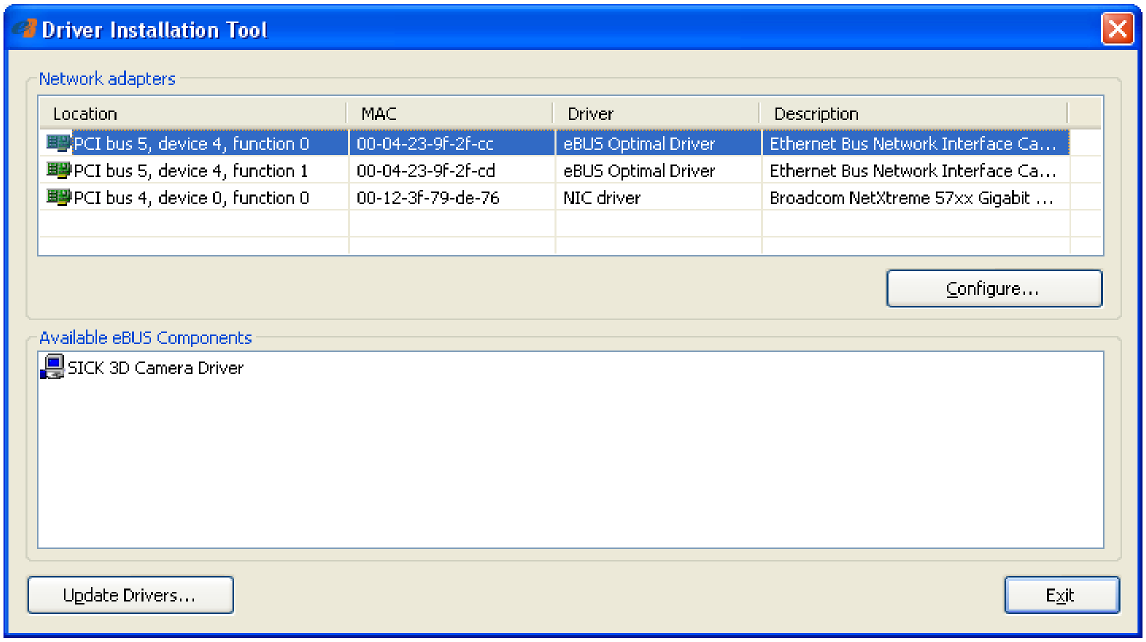
In order to optimize performance and not overload the PC with data transfer work you can use the high performance drivers by Pleora to by-pass the standard Windows IP stack. Easily described this will install a switch between the network adaptor and the IP stack which will listen for packets from a Ranger camera. These packets will be sent immediately to your Ranger application software without needing to go through the slow but general processing of the standard IP stack. This has mainly two advantages:

1. The overall CPU load will decrease which gives your application software more time to do image analysis.
2. The network data processing will be faster which reduces the risk of loosing network packets due to latency in the IP stack.

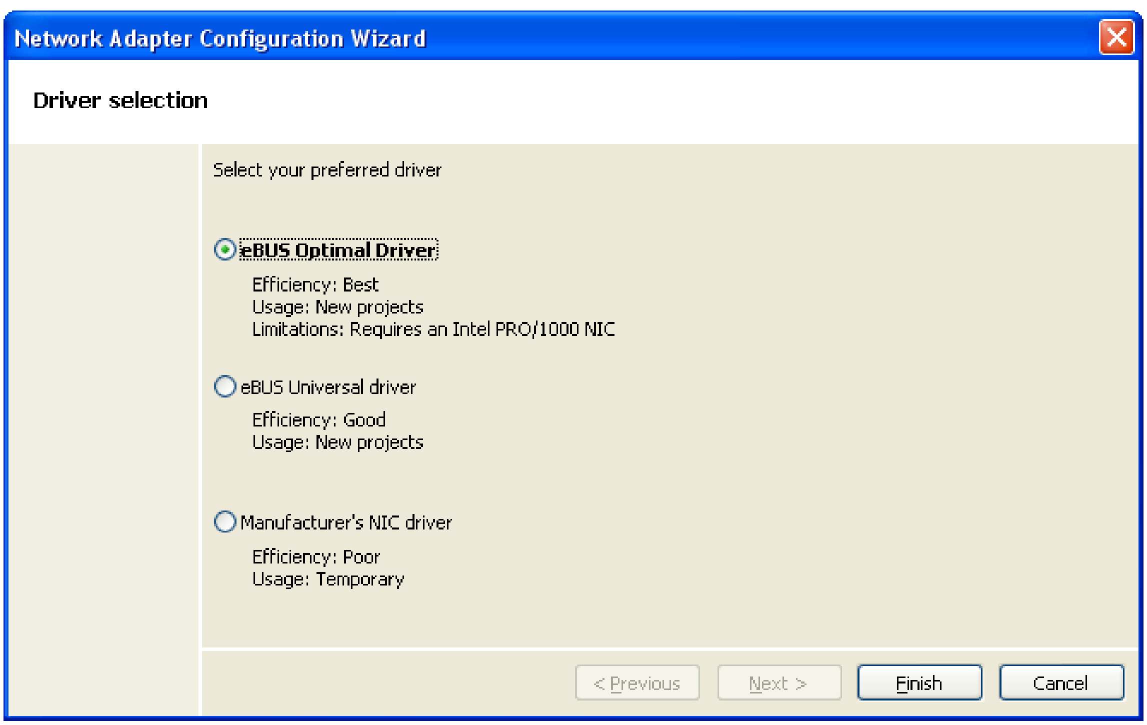
These two advantages are particularly interesting in high speed applications but may also help solving problems with lost packets.

To activate the High performance drivers you need to use the Pleora Driver Installation Tool. Note that you should **not** use the Activation Tool found in the same location. This tool is not used by the Ranger software and using it will only result in a (harmless) error message



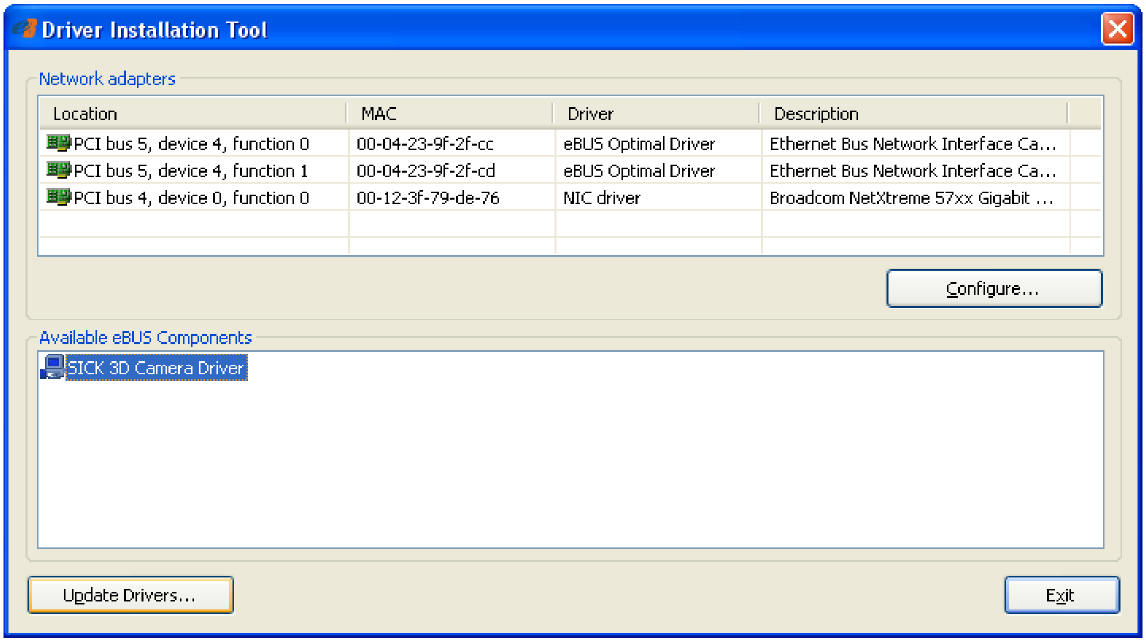


Select the network device which you wish to use to communicate with the Ranger and click 'Configure'. There are two levels of high performance drivers, Universal and Optimal. The universal driver is not as fast as the optimal but still increases performance a lot. It can be used together with any network adaptor whereas the Optimal driver can only be used together with the Intel PRO 1000 family of adaptors. If you have an Intel PRO 1000 board, select the optimal option. Otherwise, select the Universal option.

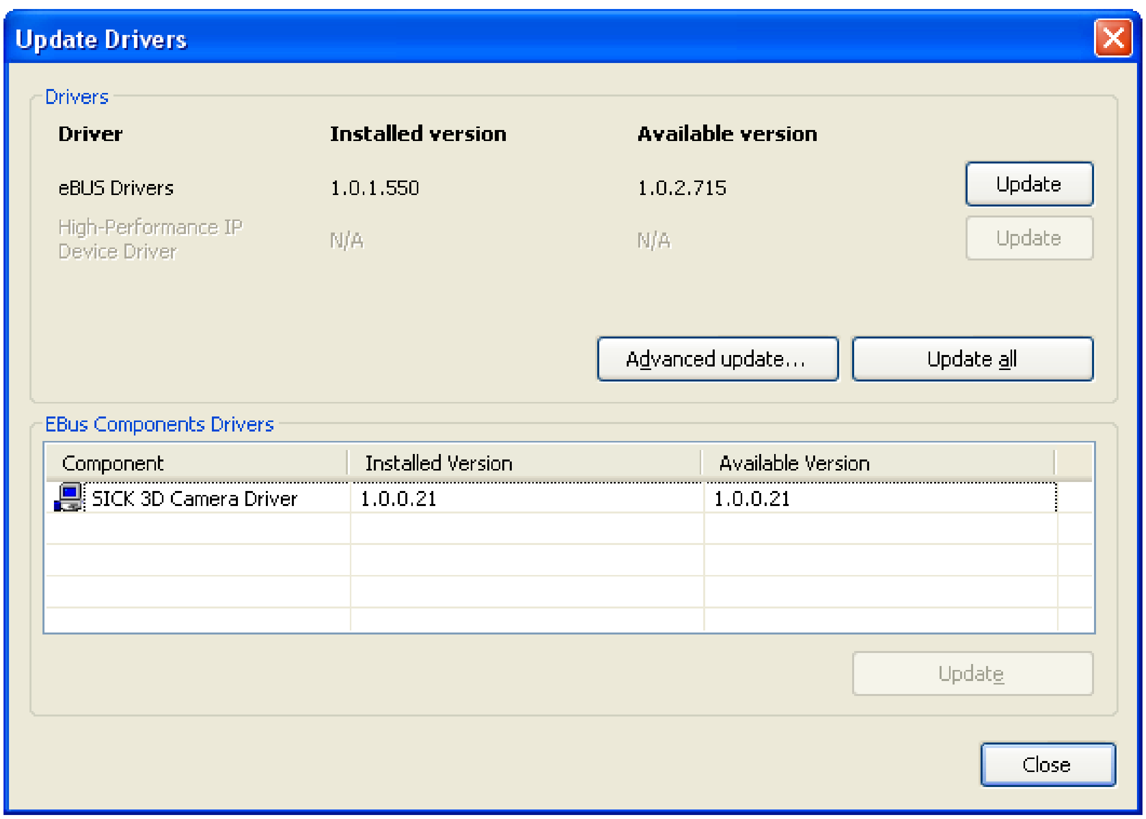


Note that if you select the optimal driver you will need to reconfigure the PC:s IP address since the high performance driver (the eBUS) shows up as a new network device. You will need to reboot the PC before the effect is seen.

Make sure the latest SICK 3D Camera Driver is used. To verify this, click on this driver in the bottom half of the configuration tool and then click Update Drivers.



If there is a newer driver available it will appear in the dialog that is now opened.



There will in that case be a yellow warning triangle on the icon of the driver and a difference in version numbers between Installed Version and Available Version. Click on the driver and then on ‘Update’ to update the driver. A reboot will be required.

After these configuration steps the High Performance drivers should be ready to use. If they work as they should, Ranger Studio will display the log message “Driver Mode: Fast” when you press Connect.

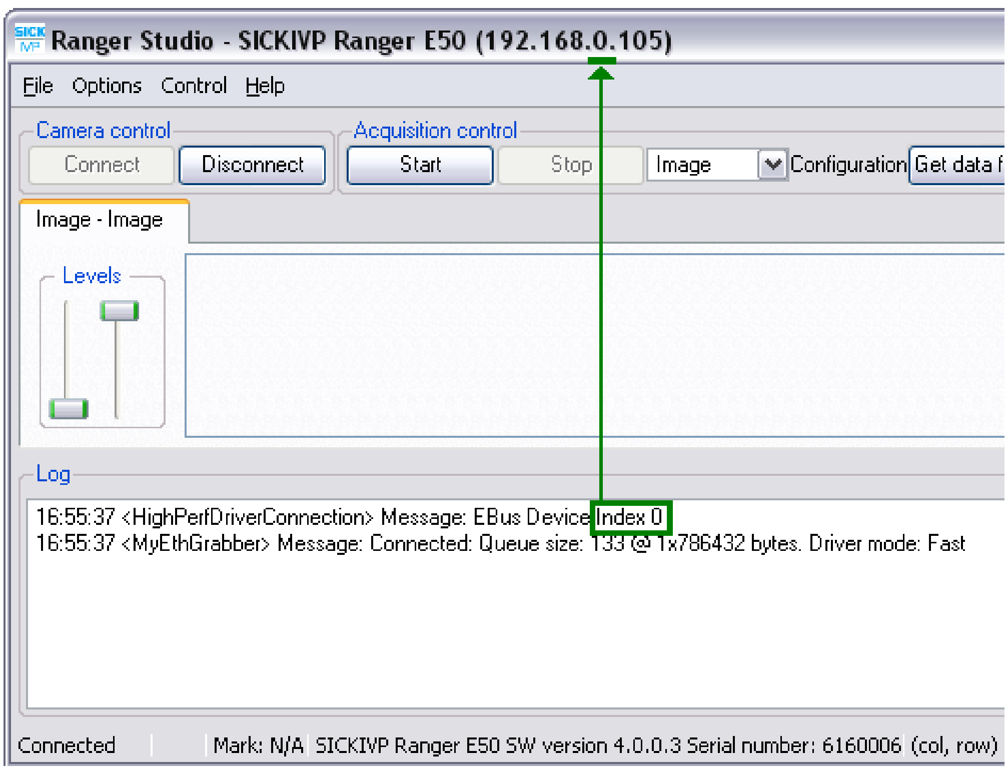
Using the high performance driver with several cameras

When using the high performance driver in a multi camera setup, it is important to configure the IP addresses so that they are mapped to the eBUS device index. If the device index is “0”, as in the image further down in this chapter, the third digit in the IP address should be set to “0” as well. For the eBUS device with index “1”, the third digit should be set to “1”, and so on.

The reason for this mapping is to reduce connection time and if this has not been done properly there is a risk that the connection times out and the normal driver is used. This is of extra importance if several cameras are used but also good to know in a one camera system. Normally connecting should take no longer than five seconds.

Mapping of IP address to eBUS device index

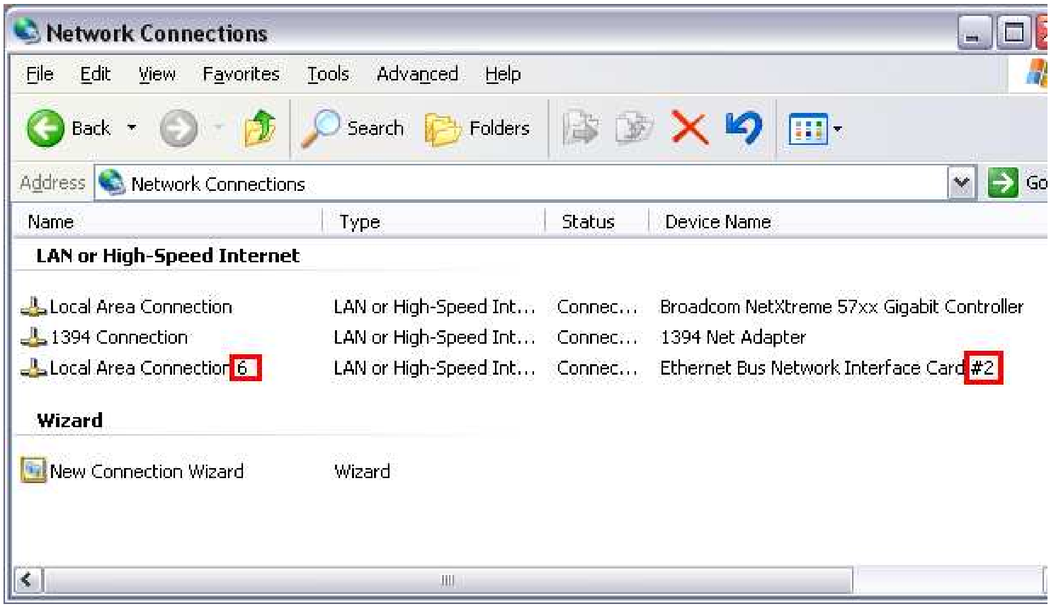
When mapping the cameras to their respective device number, first figure out the order of the eBUS devices by connecting the cameras to Ranger Studio one at a time and then read off the device index. In a good scenario and if the fast driver has been successfully installed, the eBUS device index will be printed out in the log once connected. You can then just set the IP addresses accordingly. In the below image, you can see a correct mapping where index “0” is the same as the third digit in the IP address.



Correct mapping of IP address and device number can be detected in Ranger Studio.

Sometimes the camera cannot connect through the fast driver even if it has been properly installed and instead it switches to the standard NIC driver (Driver mode: normal). The reason for this is usually that the mapping is incorrect and that connecting takes so long that it times out. This typically happens if the Pleora driver has been installed/uninstalled several times without ever being removed completely (see FAQ C16 for instructions on complete eBUS removal) and as a consequence the eBUS device index has increased significantly. Each time the fast driver is installed, the device index is, namely, incremented unless it is removed completely. In this case you need to uninstall any previous eBUS installations completely by following the instructions in FAQ C06. Once this has been done and no record of the Pleora driver can be found, it can be reinstalled and the index numbering should now start from “0”.

NOTE! The numbers found in Windows assigned to the network cards are not to be confused with the actual eBUS device index.



The red marked numbers found in Windows has nothing to do with the actual eBUS index.

Example of correct mapping in a four camera system

eBUS device index: **0**
IP address: 192.168.**0**.100
Subnet mask: 255.255.255.0

eBUS device index: **1**
IP address: 192.168.**1**.100
Subnet mask: 255.255.255.0

eBUS device index: **2**
IP address: 192.168.**2**.100
Subnet mask: 255.255.255.0

eBUS device index: **3**
IP address: 192.168.**3**.100
Subnet mask: 255.255.255.0

C03 – ICON Device Configuration and/or Ranger Studio cannot find the camera

This may have a few different explanations:

Cause	**Solution**
The camera is not powered on	Check the power supply of the camera
There is a problem with the hardware connection between camera and PC.	Check the Ethernet cables and switch if there is one.
There is another application program talking to the camera. In older versions of the Ranger PC software there was a problem with an Ethernet bus service (EBService) which could reside in memory if your application software froze or crashed for some reason.	Shut down the other application before trying to connect again. If the problem persists, make sure that you have the latest version of the ICON (Ranger) software installed on your PC. If there is a locked up EBService on your computer you will have to restart the computer for the application to start again, at least for it to start using the Pleora high performance drivers. You will notice that Windows is unable to shut down if this is the case.
The network is not properly configured.	Make sure the network device handling the communication with the camera is configured to be on the same IP subnet as the camera. For more info, read the FAQ document on how to connect a Ranger camera to a PC, [C02]
The camera has crashed. This should usually not happen but there are situations where the camera state might get out of sync with the PC. This might occur if there has been a communication problem between the two.	Power cycle the camera.
Firewall problems	The firewall might block the connect request from the application program. This is not very likely to happen and will not happen if you use the default Windows firewall settings. The remedy is to allow the application free access to the network or to disable the firewall entirely for the network port used by the camera.

C04 – Ranger Studio finds the camera but when I press Connect I get a message saying ‘Camera does not respond

See answer in C03 - ICON Device Configuration and/or Ranger Studio cannot find the camera.

C05 – I have connected to the camera but when I press Start I don't see any pictures

This may have a few different explanations:

Cause	Solution
The firewall is blocking UDP packets from the camera. Using Windows default settings this may very well occur.	Disable the firewall for that network port or make an exception for all your Ranger application programs (including Ranger Studio). We recommend disabling the firewall. Since the camera(s) should be installed on a dedicated network there should be no hostile traffic on that network which makes it ok to turn off the firewall.
The 'Use enable' parameter is turned on but no enable signal reaches the camera.	Temporarily set the Use enable parameter to 0 to let the camera scan all the time. If you are in Image mode you should check the 'Use enable' parameter of the Image mode configuration and if you are in measurement mode you should check it in the measurement mode configuration.
The 'Trig mode' parameter is set to 2 (pulse triggered) but no encoder pulses reach the camera	Temporarily set the trig mode parameter to 0 (free running) to verify that this was the problem. See the note above about configuration modes.

C06 – Do I need to use the high performance drivers?

The high performance drivers may be required if you are suffering from buffer overflow or lost packets.

There are basically two reasons why you may have problems with CPU load.

1. Your application program requires too much processing power, most likely to do its image analysis.
2. The data packet rate is high enough for the system to not quite handle the incoming UDP packets.

<u>In the case of 2</u>, you will most likely have problems with lost packets (reported on the error channel, seen in the log window of Ranger Studio). In this case the Pleora High performance drivers may be quite beneficial.

<u>In the case of 1,</u> you may benefit from using the high performance drivers but the general problem will of course not disappear. If buffers are coming in at a higher rate than what your application can handle, you will have a problem. Using the high performance drivers will speed things up a bit and perhaps this is just that little extra that makes the difference between 'too slow' and 'fast enough'. Try it but make sure to check for latency in the response and buffer overflow. How to handle buffer overflow is not a simple question. You need to figure out what requirements there are on system operation robustness and error handling. General advice:

Install the high performance drivers and try to improve performance of your image analysis schema.

Typically the CPU load when receiving data with the different drivers are the following:

- Standard: 40%
- Pleora Universal: 15%
- Pleora Optimal: <5% (often much less)

C07 – I get messages about lost packets, what should I do?

Lost packets are lost UDP datagrams from the camera to the frame grabber. They have either disappeared while being transferred over the network or they were never picked up by the receiving IP stack.

Transfer problems

If the packets are lost while being transferred there is most likely a problem with connectivity or electrical disturbances of some kind. Check the physical connection, the quality of the cables, make sure all cables are CAT 6 or at least 5e and that switches are gigabit Ethernet switches. Do not connect using 100 Mbit or 10Mbit adaptors. Only use gigabit adaptors.

Receiving problems

If the receiving PC is overloaded with work it may not be able to sort the incoming packets in time. The frame grabber will then never receive the packets and they will never end up in the designated image buffer. In this case it might help installing the high performance drivers from Pleora which are available together with the Ranger software. It might also be worthwhile looking into why the computer is overloaded. Are there other programs running? Is the image analysis schema too computationally demanding? Are the analysis and acquisition programs given a high enough priority?

C08 – What hardware requirements are there on switches and network adaptors for optimal performance? Which of the requirements can be relaxed for low and medium speed applications?

Ranger E/D

The interior HW in the camera requires a Gigabit Ethernet link. This means that whatever is connected in the other end of the (Cat5E or Cat6) Ethernet cable from the camera, it needs to be Gigabit Ethernet compatible. Else the camera will not work as expected.

This means that even if low data bandwidth is used and a 100MB link in theory could be used in terms of data bandwidth, it is still necessary (due to HW in camera) to use a Gigabit Ethernet compatible HW in the other end.

Due to the large amount of data that the Ranger delivers, it is required to connect it to the PC using a separate Gigabit Ethernet network. The performance can be further increased by using a network adapter that can send and receive Ethernet Jumbo Frames, such as the Intel PRO 1000GT Desktop adapter.

Without using jumbo frames, the maximum Ethernet frame size is 1518 bytes, which includes 46 bytes used for headers. This means that the maximum size for the data packages that the Ranger can send is 1472 bytes. When using jumbo frames the Ranger unit can support up to 4054 bytes in each data package, which corresponds to 4100 bytes per Ethernet frame.

(High performance drivers not required but needed for optimal performance.)

A gigabit switch is required even if the full speed is not used. If Ethernet Jumbo frames are to be used the switch must also have support for these. Recommended switches are:

- NetGear GS716T – 16-port switch with support for jumbo frames
- SMC SMCGS5 – 5-port switch with support for jumbo frames
- HP 1800-8G – 8-port switch with support for jumbo frames
- NSM-208G – 8-port industrial switch with support for jumbo frames

(see also FAQ H05 - What network switches are supported with Ruler/Ranger ED?)

Ruler E

The interior HW in the camera requires a Gigabit Ethernet link. This means that whatever is connected in the other end of the (Cat5E or Cat6) Ethernet cable from the camera, it needs to be Gigabit Ethernet compatible. Else the camera will not work as expected.

This means that even if low data bandwidth is used and a 100MB link in theory could be used in terms of data bandwidth, it is still necessary (due to HW in camera) to use a Gigabit Ethernet compatible HW in the other end.

Due to the large amount of data that the Ruler delivers, it is required to connect it to the PC using a separate Gigabit Ethernet network. The performance can be further increased by using a network adapter that can send and receive Ethernet Jumbo Frames, such as the Intel PRO 1000GT Desktop adapter or the 3COM 3C2000 adapter.

Without using jumbo frames, the maximum Ethernet frame size is 1518 bytes, which includes 46 bytes used for headers. This means that the maximum size for the data packages that the Ruler can send is 1472 bytes. When using jumbo frames the Ruler unit can support up to 4054 bytes in each data package, which corresponds to 4100 bytes per Ethernet frame.

(High performance drivers not required but needed for optimal performance.)

A gigabit switch is required even if the full speed is not used. If Ethernet Jumbo frames are to be used it has to be supported by the switch and the recommended switches for Ruler E are the same as for Ranger E (see previous section).

Ranger C

The following frame grabber boards are fully supported by the Ranger APIs – that is, the Ranger C can be configured and controlled, and measurement data can be retrieved by only using functions in the Ranger APIs:

- Dalsa X64-CL iPro
- Dalsa X64-CL iPro lite

In addition, any frame grabber that supports the following specifications can be used for configuring and controlling the Ranger C:

- CameraLink Base configuration with clserial dll interface version 1.1 or higher, or virtual COM port mappings.
- 115,2 kbit/s bit rate on the serial channel
- CameraLink pixel clock frequency of at least 33.3 MHz

The configuration used in Ranger C for the CameraLink serial channel is show in the following table:

Parameter	**Value**
Bit rate	115200 bps
Data bits	8
Stop bit	1
Parity	None
Flow control	None

In the initialization files for the Dalsa X64 frame grabbers, SICK IVP uses “2-taps interleaved” Camera Sensor Geometry Setting.
The important thing to know here, since the RangerC is a MultiScan camera and can generate resulting images with arbitrary image width depending on what components are enabled in the Measurement configuration, is that all data through the frame grabber is interpreted as Bytes (8 bit data) when passing through the frame grabber.

The type conversion to suitable data formats are later done in the PC buffers by the application since we know the data format and individual offsets of each component data in the total data stream from the camera.

C16 – High performance 3D camera driver is not found after SW upgrade from 3D cameras 3.1 to 3.3

When upgrading from 3D Cameras SW v3.1 to 3.3 or later, the SICK IVP 3D camera high performance driver may not be found automatically. This is due to that the default storage location has been changed in between the versions.

If you have previously had 3.1 installed and then un-installed this version (including the high performance driver) and now installed the newer 3.3 version, then you will probably see a message saying that the SICK IVP 3D Camera driver can’t be found and that you are asked to browse for this driver.

The reason for this is that WdfCoInstaller*.dll is assumed by high performance driver Pleroa 1.0.2.715, to be found in old storage location C:\SICKIVP\3d camera\drivers, even if the new

SDK is installed at the new default location C:\Program Files\SICKIVP\3d camera\drivers. The reason for this is probably that there is some old information in the registry as to where this driver can be found.

A workaround is to simply point out the new location when the program asks for this. This also goes for the Eb*.sys-file.

To solve this problem, it is better (and recommended) to after un-installation of older 3.1 version, clean the registry for any occurrence of high performance keys, before installing the new 3D Camera SW version 3.3 (or newer).

There are two ways of cleaning the registry:

How to clean the registry

1. Uninstall the existing version of Ranger Studio **and** Pleora (using “Add remove Program”).

2 Remove hidden devices (choose alternative a or b)

a) Light version: Remove hidden devices concerning eBUS (see blue text below)

b) Thorough version: Follow the instructions in the "EthernetBusTroubleshooting.pdf" (found in the installation at C:\Program Files\SICKIVP\3D Cameras\EthernetBusTroubleshooting.pdf)

3. Install Ranger Studio and new version of Pleora

Light version: Remove hidden devices concerning eBUS

Display disconnected devices in the Device Manager:

The Device Manager normally only show connected devices, and even if the option "Show hidden devices" is checked, not all installed hardware is listed within the tree view.

This is a major drawback in cases when you want to uninstall several devices (e.g. modems or other plug-and-play devices) when they are not currently connected to the computer. Normally you would start the "Add/Remove Hardware" wizard for each device you want to uninstall. However, there is a smarter solution to do this, as explained below.

SOLUTION

To show all installed devices, follow these steps:

1. Open a command prompt

2. Enter the following:

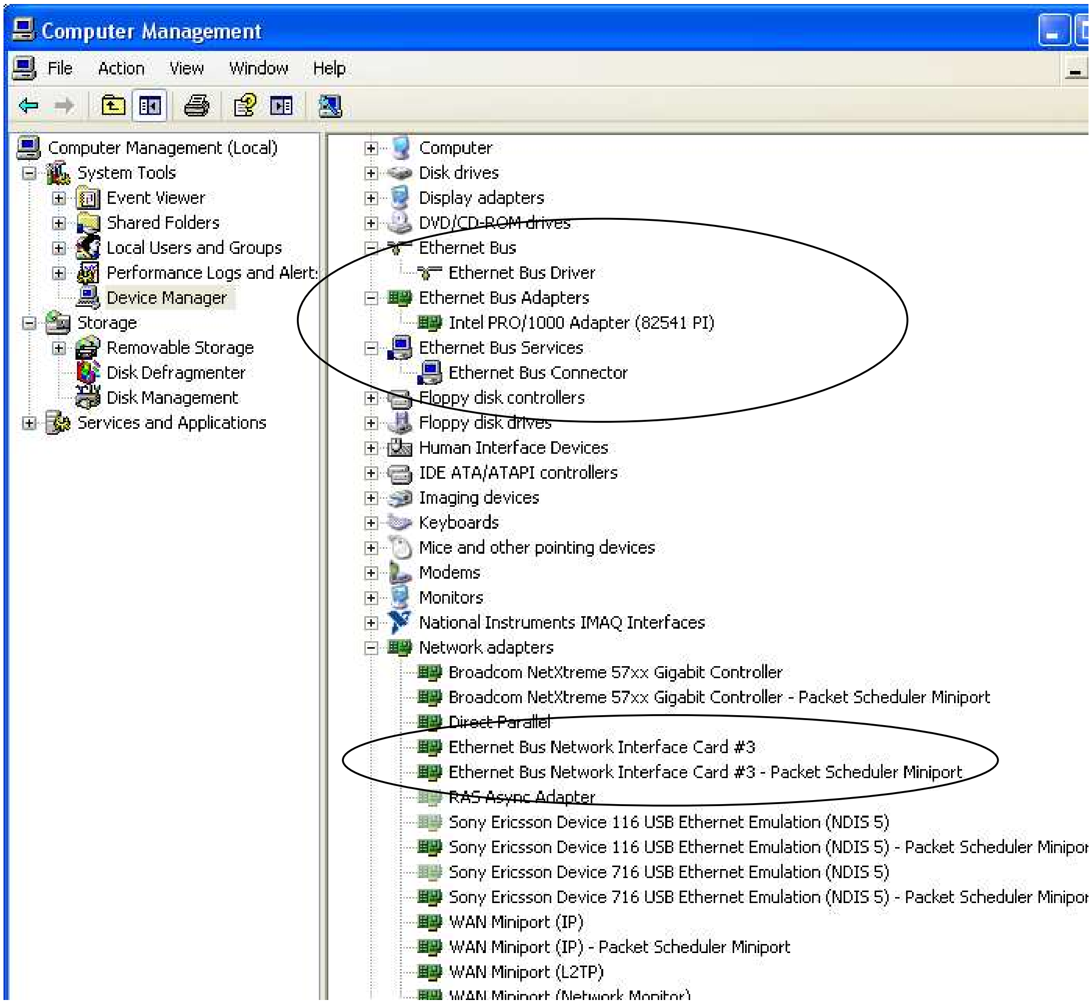
```
SET DEVMGR_SHOW_NONPRESENT_DEVICES=1
cd %systemroot%\system32
start compmgmt.msc
```

3. Select the Device Manager node and turn on the "Show Hidden Devices" feature by clicking the appropriate entry in the View menu.

The Device Manager will now list all devices that are installed on the machine.

Uninstall all found eBUS devices.

Example below from the Computer Management:



Computer Management
File
Action
View
Window
Help
Computer Management (Local)
System Tools
Event Viewer
Shared Folders
Local Users and Groups
Performance Logs and Alert:
Device Manager
Storage
Removable Storage
Disk Defragmenter
Disk Management
Services and Applications
Computer
Disk drives
Display adapters
DVD/CD-ROM drives
Ethernet Bus
Ethernet Bus Driver
Ethernet Bus Adapters
Intel PRO/1000 Adapter (82541 PI)
Ethernet Bus Services
Ethernet Bus Connector
Floppy disk controllers
Floppy disk drives
Human Interface Devices
IDE ATA/ATAPI controllers
Imaging devices
Keyboards
Mice and other pointing devices
Modems
Monitors
National Instruments IMAQ Interfaces
Network adapters
Broadcom NetXtreme 57xx Gigabit Controller
Broadcom NetXtreme 57xx Gigabit Controller - Packet Scheduler Miniport
Direct Parallel
Ethernet Bus Network Interface Card #3
Ethernet Bus Network Interface Card #3 - Packet Scheduler Miniport
RAS Async Adapter
Sony Ericsson Device 116 USB Ethernet Emulation (NDIS 5)
Sony Ericsson Device 116 USB Ethernet Emulation (NDIS 5) - Packet Scheduler Minipor
Sony Ericsson Device 716 USB Ethernet Emulation (NDIS 5)
Sony Ericsson Device 716 USB Ethernet Emulation (NDIS 5) - Packet Scheduler Minipor
WAN Miniport (IP)
WAN Miniport (IP) - Packet Scheduler Miniport
WAN Miniport (L2TP)

Data Quality and Optimization

D05 – Exposure is uneven when I trig the camera with encoder. How can I get the same exposure time for every profile?

This is most likely due to using *profile triggering* and at the same time running the camera in *implicit reset* mode which causes the exposure time to be conveyor speed dependent. (If you need more information on implicit & explicit reset, please read FAQ entry O06)

There are basically three ways to come around this problem:

1. **Use explicit reset mode**
   - **Pros:** Exposure time is always constant
   - **Cons:** Lower maximum scan rate
2. **Flash the laser (not currently supported on Ruler cameras)**
   - **Pros:** Allows for high scan speeds, prolongs laser lifespan
   - **Cons:** Sensitive to external light, normally requires higher laser power
3. **Use Mark functionality to measure inter profile distances**
   - **Pros:** Exposure time is constant, allows for high scan speeds
   - **Cons:** profile sampling distance not constant, data processing becomes harder

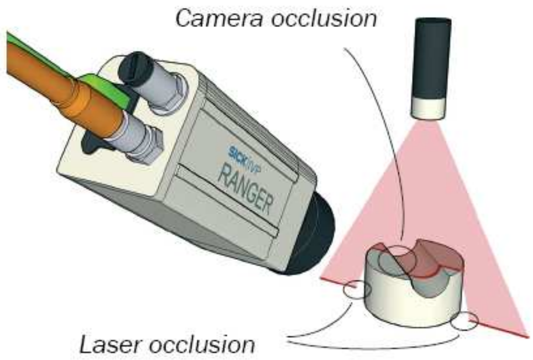
D07 – I have problems with missing data, what should I do?

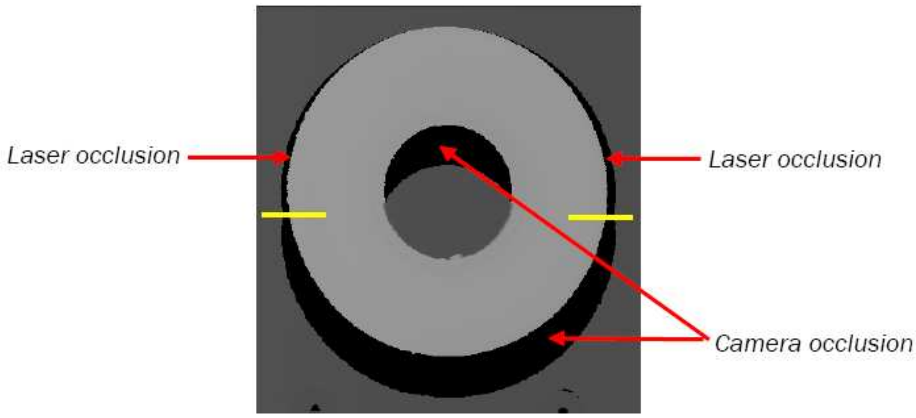
Missing data are regions in the range image containing no information (having the pixel value 0). Missing data can either be caused by occlusion or by the intensity of the reflected laser light being below the threshold(s) of the 3D algorithm. The last case is most often due to the sensor being underexposed (or thresholds being set too high).

Occlusion

There are two kinds of occlusion in a laser triangulation system:

- Camera Occlusion – The laser line is hidden from the camera behind object features
- Laser Occlusion – The laser cannot illuminate parts behind object features





Laser & Camera Occlusion

To reduce the occlusion you can:

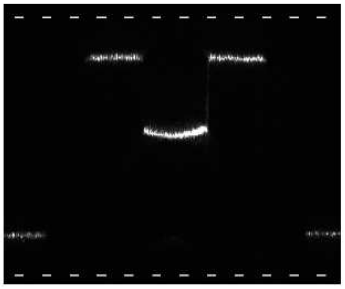
1. Reduce the view angle between the camera and the laser (this has a negative effect on the range resolution).
2. Use an additional camera to look from the opposite side of the object (increases the price of the solution).
3. Use a multiscan setup with two triangulation lasers illuminating the object from different angles (less costly than adding an additional camera, but reduces the maximum scan speed)
4. Use a laser with a low fan angle and place it further away from the object (reduces laser occlusion, but increases the physical size of the setup)

Underexposure

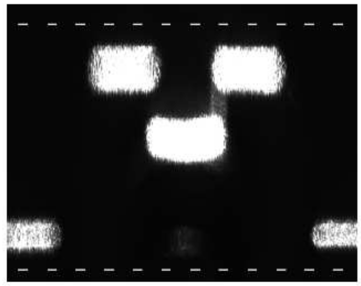
Any sensor column containing no pixels with values above the threshold level(s) of the 3D algorithm will give a missing data pixel in the 3D image. Missing data can appear either due to too high threshold settings or due to underexposure of the sensor (not enough light is reflected back on the sensor).

The first thing to do if you have problems with missing data not caused by occlusion is to have a look in image mode configuration. First make sure that you use the same exposure time & gain settings as you do in your 3D component to be able to compare the results.
Make sure you have good focused camera lens and also good focused laser.
Then adjust the aperture, exposure time & gain until the laser line is correctly exposed; it should neither be too dark nor too bright. Try to minimize the number of saturated pixels, i.e.

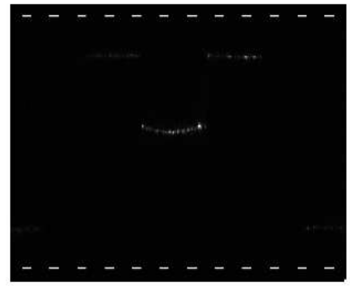
aim at gray scale values < 255.



Correct laser line exposure.



Overexposed laser line.



Underexposed laser line.

Tip: In Ranger Studio you can mouse over the image to see individual pixel values in the status row at the bottom of the main window.

When you are happy with the exposure of your well focused laser line don't forget to transfer the settings used in image mode (gain & exposure time) to your measurement component.

Now it's time to adjust the threshold levels of your 3D component. You can actually do this "on the fly", i.e. when the camera is running. Switch to measurement mode, start the camera and decrease the threshold values until the amount of missing data is acceptable. Be careful not to decrease the thresholds too much as this could increase the noise in the range image instead.

D08 – Is there a band pass filter in the Ruler? What are the characteristics of the BP-filter?

Yes there is an in-built band pass filter inside the Ruler mounted in front of the sensor. Both RulerE and older versions of Ruler (30/100) use the same sort of BP filter. The purpose of the filter is to reduce the impact of ambient light in the images. The filter is 60 nm FWHM and is adjust to fit the wave length of the laser 660 nm +/-15nm.

D09 – How come I'm getting noisy range data when my conveyor is running at low or moderate speeds?

See answer in "D05 - Exposure is uneven when I trig the camera with encoder". How can I get the same exposure time for every profile?

Hardware and Accessories

H03 – What network adapters are recommended for the Ethernet cameras

The main network adapter requirement is that it needs to support a speed of 1Gbit/s. For certain configurations support for Jumbo frames can help optimize performance, so this is recommended, but not mandatory.

In SICK 3D Camera SDK releases prior to 4.3, certain Intel network adapters were required in order to use the high performance driver called 'eBUS Optimal' – the driver which offered best performance. Today, after the transition to eBUS 2.0, only one high performance driver exists for SICK 3D Camera SDK: 'Universal Pro'. This driver works with any adapter that fulfils the basic requirements.

Note! It is still possible to choose eBUS Optimal during installation, but it will not work with SICK's 3D cameras. This is a driver intended for other 'GigE Vision' devices.

Even if Universal Pro supports a wide range of network cards, we have continued using some specific Intel adapters when testing and developing our cameras. The reason for this is mainly because they have proven to be stable, but it also gives us a number of adapters we can recommend with confidence.

Below follows a list of the network adapters we test frequently and thus recommend:

- Intel® Gigabit CT Desktop adapter
- Intel® PRO/1000 GT Desktop adapter
- Intel® Gigabit ET Dual Port Server
- Intel® PRO/1000 PT Quad Port Server Adapter

H04 – What frame grabbers are supported with Ranger C?

The information about supported frame grabbers presented below can be found in the Ranger C Operation Instructions in appendix C.

The following frame grabber boards are fully supported by the Ranger APIs – that is, the Ranger C can be configured and controlled, and measurement data can be retrieved by only using functions in the Ranger APIs.

- Dalsa Coreco X64-CL iPro (PCI card. Driver is found on SICK IVP support pages)
- Dalsa Coreco X64-CL iPro lite (PCI card. Driver is found on SICK IVP support pages)

Also, there rest of the Dalsa X64 family should work in theory but no release tests have been performed by SICK IVP to guarantee the full functionality. SICK IVP only claims to fully support the iPro family.
Other Dalsa Coreco FGs are:

- Dalsa X64-CL (PCI card)
- Dalsa X64-CL Dual (PCI card)

In addition, any frame grabber that supports the following specifications can be used for configuring and controlling the Ranger C:

- CameraLink Base configuration with clserial dll interface version 1.1 or higher, or COM-port mappings
- 115,2 kbit/s baudrate on the serial channel
- CameraLink pixel clock frequency of at least 33.3 MHz

Frame grabbers from the following manufacturers have been verified to work:

- Active Silicon
- Euresys
- Matrix Vision
- Matrox (Helios and Solios families recommended, we don't recommend using Oddysey FG since the serial drivers are not optimized for high baudrates for that FG and we have seen communication problems with the RangerC camera)
- National Instruments (NI 1428 PCI)
- Seldes

H05 – What network switches are supported with Ruler/Ranger ED?

There are some features of a switch which are important to consider when choosing a switch for connecting several rangers to one PC. The first thing is that the switch must be a gigabit switch. Another thing is if the switch is capable of handling Ethernet Jumbo frames or not (Jumbo frames are not required but needed for optimal performance).

The following two switches are gigabit switches that handle Ethernet Jumbo frames.

- NetGear GS716T - 16-port switch
- SMC SMCGS5 – 5-port switch

The switches in the list below have been confirmed by users to be working with our cameras. This list will be updated continuously.

- Linksys SD 2008

(this same info is also available in FAQ C08 - What hardware requirements are there on switches and network adaptors for optimal performance? Which of the requirements can be relaxed for low and medium speed applications?)

H06 - How do I create a calibration target suitable for my setup?

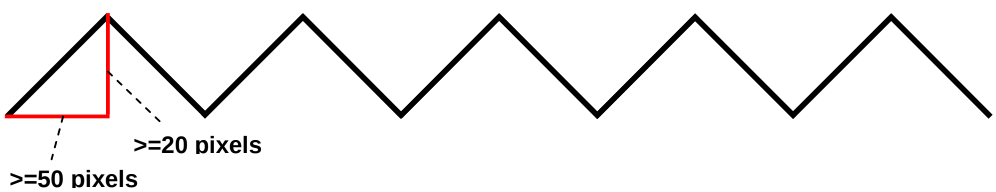
Size of the target

There is no theoretical limitation in the calibration target size and as long as the sensor image of the laser line is good enough, calibration should be feasible. However, for smaller targets the quality requirements are greater than for bigger ones (thinner laser line, more even surface, etc.).

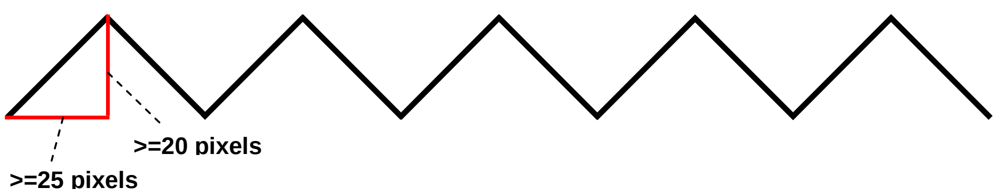
For height calibration into real world coordinates the saw teeth side is used. Here at least 4 saw-teeth have to be visible in the FOV all the time and a certain amount of sampled points has to be collected on each tooth side in order for a calibration to be feasible. The requirements for the number of sampled points vary with the FOV width chosen. If for instance 1536 pixels are used when calibrating (full sensor width of a Ranger D/E 50/55), each tooth side must be sampled by 50 points in the x-direction. If half the sensor is utilized, 768 pixels, only 25 sample points are needed. So, the number of sample points required is directly proportional to the number of sensor pixels used. Consequently, this means that the quality restraint is loosened up as well, since fever sample points means less accuracy.

Unlike the width requirement, the teeth are not scaled height-wise and these always require at least 20 pixels on the sensor. Below you will find an illustration of how the number of pixels employed on the sensor affects the calibration target properties.

Requirements for FOV width 1536 pixels

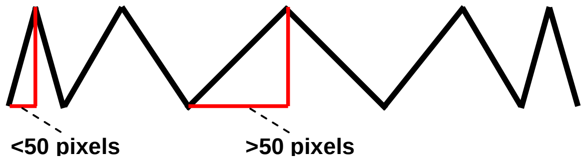


Requirements for FOV width 768 pixels



The FOV used is in other words something that should be taken into consideration before a calibration target is manufactured – the same camera and geometry may require different calibration targets if the number of pixels employed on the sensor is to be changed. Wide angled optics is another pitfall that can render a target useless. A target may meet the requirements in the middle of the FOV, but towards the sides lens distortion will make the teeth sides appear thinner. This is illustrated in the image below.

Geometry and lens distortion affecting saw teeth width



The 20 pixels in height for each tooth is perhaps of extra importance and if the height covers fewer pixels, the saw teeth side may be interpreted as flat(!). This can ruin the calibration of the lens and perspective distortion.

Example: What size must my calibration target be, given the following properties?

Width resolution of the sensor: 1536 pixels
FOV width: 300mm
Pixel resolution: 300mm/1536pixels≈0.2mm/pixel

Maximum target proportions

We know that at least 4 peaks have to be visible on the sensor at all times. The biggest allowed saw tooth width will thus be:

300mm/4=75mm

Height with 90° angle:

37,5mm

Minimum target proportions

One saw tooth must at least cover 50pixels*2=100 pixels in width. One peak must thus be at least:

0.2mm/pixel*100pixels=20mm

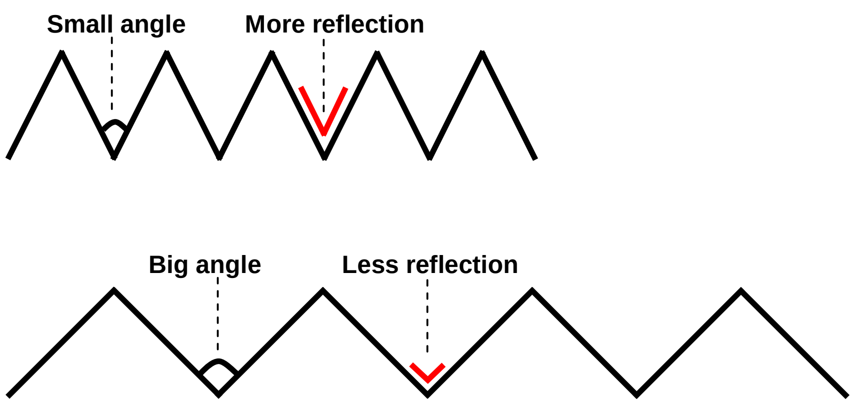
Height with 90° angle:

10mm

NOTE! Both these measurements are at the very brink of what is possible <u>in theory</u>. It is thus NOT recommended having a calibration target with a size of the above calculated values. With a full sensor (width 1536 pixels) the recommended number of teeth is somewhere between 6 and 8. For an E40 camera on the other hand (width 512 pixels), it is better to aim at 5-6 teeth. Depending on the geometry, these recommendations will not likely be fulfilled on the entire sensor, it is merely a recommendation for the middle part height-wise.

Shape of the calibration target

As long as the above stated criteria are fulfilled, the calibration process should work fine. However, it may be difficult to use a calibration target of "extreme" shape, such as pointy teeth. The smaller the angle, the more light will be reflected back in the calibration target itself. This makes the calibration process more difficult and the surface of the calibration target has to be less reflective. The general recommendation is to have an angle of at least 90 degrees between two teeth, as in the bottom part of the image below. Not that this is only a recommendation and sometimes other target shapes are preferred.



Material properties

In the previous section (Shape of the calibration target) we discussed how the saw-teeth angle affects the amount of light reflected back upon the calibration target itself. An adding factor to these reflections is bad material characteristics and therefore we want a surface that scatters the laser beam as little as possible.

There are a couple of things one may want to take into consideration when developing a calibration target in regards to the material properties. If the target is to be used in a big geometrical setup the smoothness of the target surface is not as vital as for small calibration objects. Two calibration targets of different sizes, but with the same surface properties, will not be equally affected from a percentage aspect if the same laser is used. The general rule of thumb is that the smaller the object is the smoother the surface has to be (and if much smaller the laser line has to be thinner as well).

The recommended material for calibration targets is anodized aluminium which has a gray smooth surface, given that the milling process has been adequate.

H07 - How do I maximize laser lifetime in Ruler E 3B?

The expected lifetime of a laser module is more or less defined by the time the laser is powered on and lit. Factors that affect the actual time is the ambient temperature and the laser output power. Lasers thus behave much like overhead projectors. Below are some suggestions on how to maximize the laser lifetime by minimizing the on time.

Methods of turning the laser off

The Ruler has a number of built in functions to limit the laser on time. Below is a brief summary. Each of the points will be addressed in detail later.

Laser power connector	Use this for safe maintenance
The “Laser On” parameter	Use this to turn the laser off during lunch breaks etc.
The “Enable” signal	Use this to turn the laser off while there are no objects in the field of view.
Laser pulsed by the acquisition	Use this to minimize laser usage during the acquisition cycle

Laser Power Connector

The laser is powered by pin 8 of the Power connector while the rest of the Ruler is power through pin 2. This means that a simple way to shorten the on time of the laser is to simply cut its power. This will however cut the power of the whole laser module. This module also contains electronics to control the laser diode power and it is therefore not recommended to use this method for frequent cutting of laser power. Frequent power cycling of electronics generally increases the risk of damage due to electric perturbations. It is however a good way to ensure that the laser is disarmed when performing maintenance on the machine.

The “Laser On” Parameter

Ruler parameter files contain a “Laser On” parameter in the “System” section. Setting this parameter to “0” will turn the laser off while “1” turns it on. This parameter is connected to the trig input of the laser module and does not cut the power of the laser module itself. Only the laser diode is turned off which means this method can be used as frequently as desired. There are two ways of accessing this parameter.

1. By changing it in the parameter file.
2. By using the setParameterValue() member of the Icon::Camera class. This is only applicable if you use iCon to control the Ruler. If you use the Ruler API you always need to upload a parameter file when you wish to change a parameter.

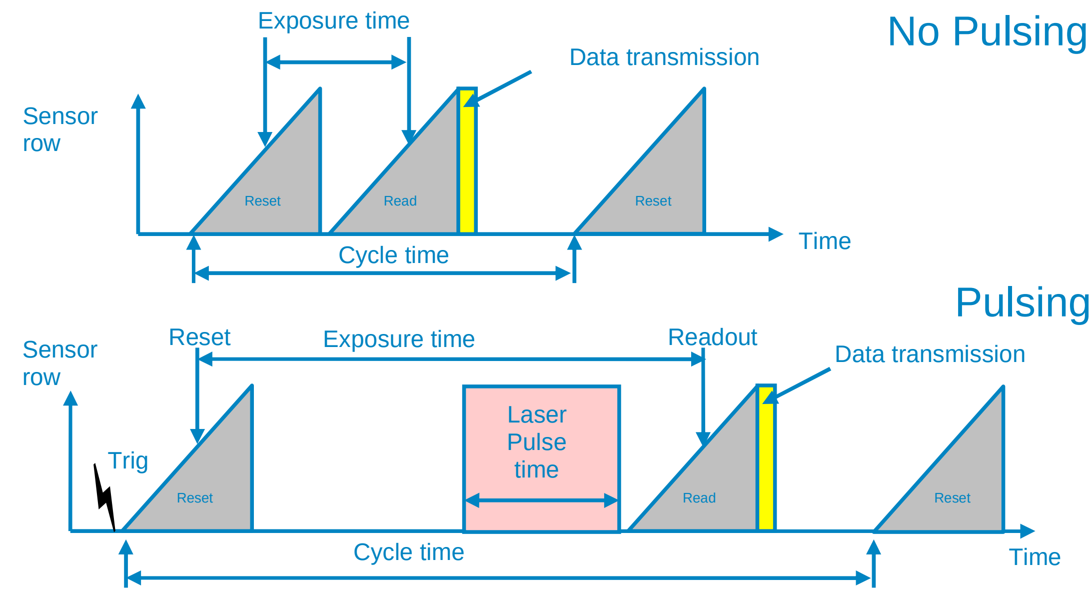
The “Enable” Signal

The Ruler has an input pin named “In 1” (pin 1 of the Power connector). This 24 V input is typically used to trigger acquisition of an image buffer (a sequence of scans). It can however also be configured to control the laser. If the “Laser On” parameter described above is set to “2”, the laser will be lit only when the camera is in RUN mode and when the enable input is high (24 V). By using e.g. one or a couple of photo switches in combination with this function the laser on time can be reduced dramatically and automatically.

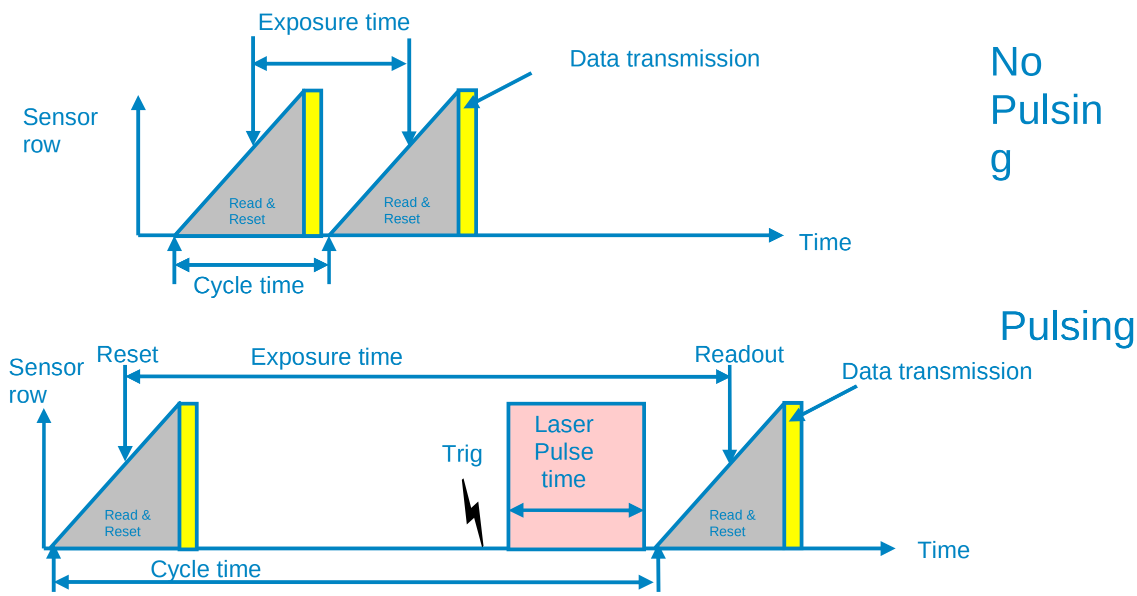
Laser Pulsed by the Acquisition

The final method of reducing laser on time is by pulsing of the laser. This basically means that a short pulse of laser light is emitted while the sensor is collecting photons. You activate this function by setting the value of the “Laser pulse time” parameter of the 3D component you use (i.e. Hi3D or HorThr) to something other than 0. Note that setting this parameter means increasing the minimum cycle time by up to the same amount. This means that the camera might be slower since it needs the time to pulse the laser. If this is the case or not depends on your cycle time and exposure time settings.

The figures below show graphically where the laser pulse time is introduced. First in explicit reset mode (cycle time > exposure time) and then in implicit reset mode (cycle time < exposure time).



Explicit Reset mode and laser pulsing. *The top graph shows the minimum cycle and exposure time without pulsing. The bottom graph shows where the laser pulse time is introduced. Note that in this case the exposure time is set much larger than the minimum exposure time. In that case, laser pulsing will not affect the minimum cycle time. Adding laser pulsing to the scenario described in the top graph means squeezing the pulse in between the reset and read ramps which means increasing the minimum cycle time.*



Implicit Reset mode and laser pulsing. *Here the minimum cycle time is always increased by the laser pulse time. But on the other hand it provides a good way of obtaining a controlled exposure time when using external triggering. It also provides a way to avoid the problem with over exposed images when using the full sensor for range data. The minimum exposure time* *can otherwise be a problem in that scenario.*

Programming using ICON API

I03 – How many image buffers can I use on the PC side?

The number of image buffers allocated by the frame grabber is determined by the following factors:

1. The number of Mega bytes assigned to use for image buffering (frame grabber parameter).
   This size is then the global size allocated, later split up in between all image buffers.

2. The number of rows in a single image buffer (frame grabber parameter).
   This number and the data format from the current configuration in the camera decides the resulting number of possible buffers to hold the received data from the camera.

3. The total size (in bytes) of each scan (consequence of the chosen data format).
   This is determined by the current camera configuration.

The first two are simple parameters that you set when you initiate the frame grabber objects. Number 1 is of course also limited by the amount of available RAM in your computer.

The last one is determined by the data format which results from the parameters of the currently active configuration. A few examples:

1. A single Hi3D algorithm configuration using all 1536 columns is 2x1536 bytes of range data + 1x1536 bytes of intensity data = 4608 bytes per scan.
2. A Multi Scan setup using one Hi3D component (4608 bytes), one HighRes Gray (1x3072 bytes) and one Scatter (1x1536 bytes of scatter direct + 1x1536 bytes of scatter side-band) = 10752 bytes per scan.

The total number of allocated buffers is simply as many buffers as fit into the assigned memory. Example 1 above: using e.g. 400 rows per buffer results in 1 843 200 bytes per buffer. Using 25 Megabytes for buffering means 25 * 1024 * 1024 / 1 843 200 = 14.2, meaning 14 buffers are allocated.

I04 – How many rows are there in an image buffer? How does it relate to the enable signal and Scan Height parameters?

The number of rows in an image buffer on the PC side is determined by the frame grabber settings.
It has thus nothing to do with how many scans the camera acquires or what the Scan Height parameter is set to.

When you connect the frame grabber object to the camera you determine how many rows should go in each image buffer. This is set with the Number of Rows parameter in the frame grabber options in Ranger Studio.

The Number of Rows, together with the size in bytes of each profile and the number of megabytes to use for image buffering determines how many image buffers the frame grabber can allocate. Whenever a camera acquires a profile it is sent directly to the frame grabber which stores it on the next available row in the currently active image buffer in the PC. Whenever the number of rows of that image buffer is exceeded, the image buffer is marked as full and the next image buffer becomes active instead.

The Scan Height parameter on the camera side determines how many scans should be acquired, and sent to the PC, for each time the enable signal is activated (positive edge). It is thus only used if the 'Use enable' parameter is set to 1.

In practise you almost always want to set the Scan Height and Number of Rows parameters to the same value. Anything else (if you don't know exactly what you are doing) would probably result in non-synchronized images in the PC buffer compared to what you expect when the Enable trig signal goes high.

I08 – Should I use polling or callback to obtain data

There are two ways to obtain data from the Ranger and Ruler cameras. By using polling or callback functions. By using polling the user application program ask ICON if it has any new buffers available. It will then return the next available buffer if there is one. If there is a queue of buffers waiting you will get the first one that arrived in the queue. Using callbacks means that ICON calls a function specified by the user application program whenever a new buffer is received. If there is a queue, that is, if a previous callback function is still executing, ICON will wait and call back as soon as the first callback terminates.

The easiest method to use is polling. You ask ICON for new buffers and if a new buffer is available it is returned to you and you can do your analysis on it. There is no need to worry about multi-threading or synchronization between different components of your program. The benefit of callbacks is that especially when using several cameras, multi-threading makes it easier or at least more flexible to let different software modules operate as soon as their required information is available. If you use polling and several cameras the cameras risk getting stuck waiting for each other. Imagine for instance that camera 1 generates images 1.5 times as fast as camera 2. If you do a simple loop grabbing from cam1 and then cam2, you will eventually run out of buffers for cam1. It is possible to use polling in this case too. You can e.g. set a very short timeout for the grab functions and skip the part analysing the response from cameras than have not yet sent any images. So, in short: polling is easier to implement and understand but callbacks are more flexible when several cameras are connected to the same PC.

NOTE!
In RangerC/D/E SW version 3.3.1.1 and newer versions, it is not possible to use polling with auto release of the buffers when using the high performance driver. For this you have to specify manual release with setReleaseType and use releaseIconBuffer for releasing the buffers.

The Pleora High Performance driver can be installed in two different versions; Optimal and Universal. To install the Optimal version an Intel gigabit network card is needed.

I09 – When is a scan sent from the camera to the PC? Where is it stored?

As soon as a scan is acquired by the camera it is directly sent to the frame grabber object and stored in a PC buffer. The user will get access to the PC buffer when the buffer is filled. The number of scans in measurement mode to fill up the resulting PC buffer is specified by the user.

I10 – Can I access resulting data buffers in an arbitrary order?

It is not possible to access iconBuffers from the frame grabber in an arbitrary order.
It is always the next buffer in turn that you have access to. It is however possible to have access to several buffers at the same time after retrieving them from the frame grabber and copy them to other place and later choosing to process them in an arbitrary order.

I11 – Is the data from the Ranger calibrated mm values?

No, the data from the Ranger is sensor data and not calibrated mm data. To receive calibrated mm data you will need to calibrate the system.

I17 – How is the Ruler calibration performed?

Each Ruler is calibrated at production. A reference object is used to collect a set of points, each point having known sensor and real world coordinates. After the data collection, unknown points are interpolated from the initial set to create a LUT (Look Up Table) used to transform sensor coordinates into real world coordinates.

The LUT is stored in the Ruler flash memory and uploaded to the PC where the run-time calibration is performed. The run-time calibration consists of a transformation from sensor coordinates to real world coordinates and alternatively a rectification (re-sampling) of the points to an equidistantly sampled grid.

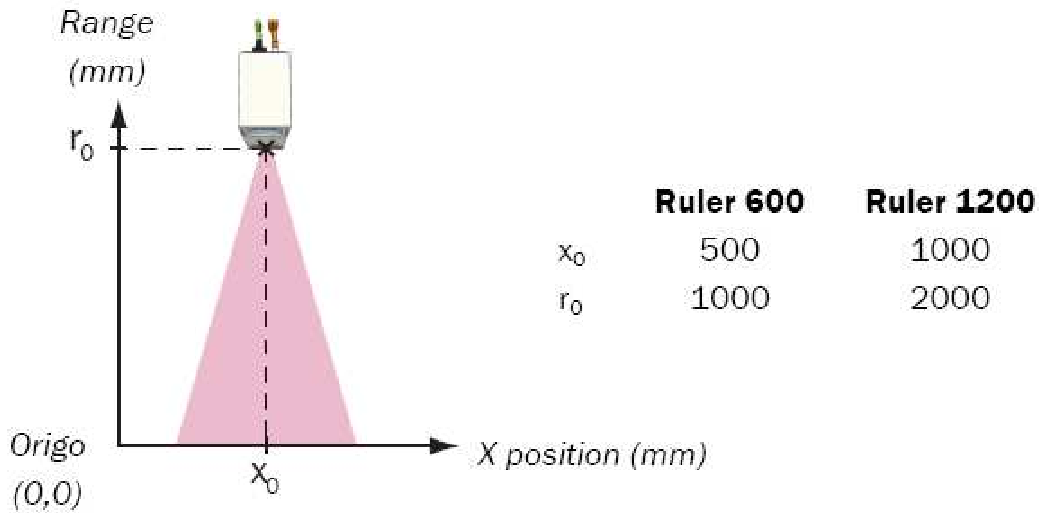
The Ruler API function `requestDataSeparate` returns calibrated *x* and *range* data. To rectify the measurement values, set the `RectifyWidth (using the API` function `setRectifyWidth`) property to the desired number of x values in the resulting profiles. If set to `0`, the measurement values will not be rectified.

Please note that rectifying the measurement values will add extra load on the CPU. *For more information please refer to the Ruler Reference Manual pp. 47.*

I18 – How can I align my Ruler to an external coordinate system?

NB: This is an excerpt from the Ruler Reference Manual pp.44-45

The measurement values that are retrieved from the Ruler are always calibrated to a local coordinate system. This coordinate system is positioned according to the figure below.



Location of the Ruler in the local coordinate system.

By setting a transformation for the Ruler object, you can change the coordinate system and receive the measurements in a coordinate system that suits your environment. This is particularly useful if you have several Rulers that measure the same object.

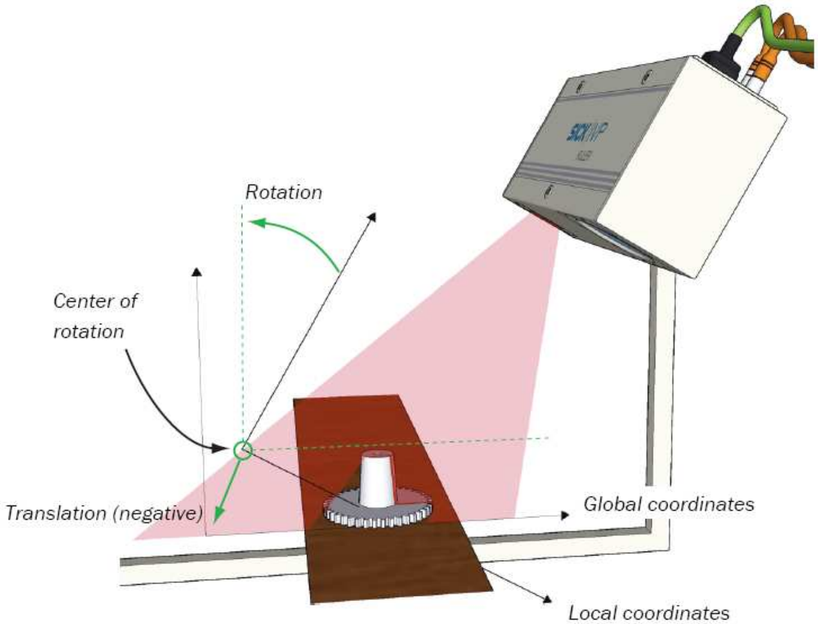
You set a transformation by setting the `Transformation` property of the Ruler object. The transformation consists of two parts:

1. `Transformation.Rotation`
   A counter-clockwise rotation, specified by the angle in radians (`Angle`) and a center of rotation (`Center`), which is a point in the Ruler's coordinate system

2. `Transformation.Translation`
   The distance to move the origin of the Ruler's coordinate system along the x axis (`x`) and the z axis (`range`).

When you have set a transformation, the Ruler object must be updated for the translation to take effect.

Note that the transformation is performed on the PC and not in the Ruler. Therefore it is included in the parameter files, but must be set by your application. If you always want to use the same transformation with the Ruler, your application should save the transformation parameters in order to be able to retrieve them when re-initializing the Ruler.



Transforming the coordinate system

I19 – How can I align several Ruler cameras?

See answer in FAQ I18 above.

I20 – Can the Ranger software be used together with Windows Vista?

Windows Vista is currently not supported but will be supported in the future.

I21 – How do I convert VS2003 project files to VS2005?

Let VS2005 automatically do the conversion.

Make sure that for Debug mode:
Configuration Properties - C/C++ - Code Generation - Runtime Library =
= Multi-threaded Debug DLL (/MDd)
and Configuration Properties - Linker - Input - Additional Dependencies = icon_vc80d.lib.

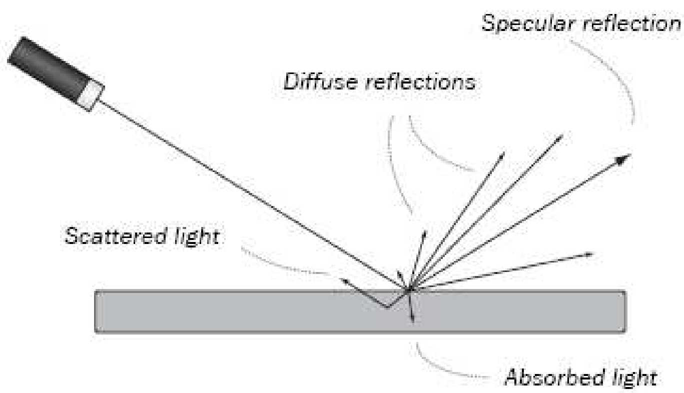
For Release mode:
Configuration Properties - C/C++ - Code Generation - Runtime Library =
= Multi-threaded DLL (/MD)
and Configuration Properties - Linker - Input - Additional Dependencies = icon_vc80.lib

Laser Triangulation

L01 – How do I choose camera/laser geometry?

Choosing the right way of mounting the Ranger and the illumination is often crucial for the result of the measurement. Different geometries have different pros and cons and there is not one setup that is always going to be the best choice. Which geometry to use depends on a number of factors, for example what is going to be measured, surface characteristics, shape of the object, resolution and speed requirements.

Measuring with the Ranger means measuring light that is reflected by objects, and from these measurements draw conclusions of certain properties of the objects. For a machine vision application to be efficient and robust, it is therefore important to measure the right type of light. The image below shows different types of reflections that exist.



Surface characteristics of the object

Different object surfaces reflect light in different ways and we need to have a camera/illumination setup where the light that is to be measured reaches the camera. Matte objects reflect light in a very different way compared to glossy objects. On glossy surfaces, all light is reflected with the same angle as the incoming light, measured from the normal of the surface. This is called the *specular* or *direct reflection*. Matte surfaces reflect the light in many different directions. Light reflected in any other direction than the specular reflection is called *diffuse reflection*.

On some materials, the light may also penetrate the surface and travel into the object, and then emerges out of the object again some distance away from where it entered. If such a surface is illuminated for example with a laser, it appears as if the object “glows” around the laser spot. This phenomenon is used when measuring scatter. The amount and direction of the scattered light depends on the material of the object.

Different objects absorb light with different wavelengths differently. This can for instance be used for measuring colour or IR properties of object. The amount of light that is absorbed usually decreases as the incoming light becomes parallel with the surface. For certain angles, almost all light will be reflected regardless of wavelength. This phenomenon is used when measuring gloss, which can be used for example for detecting surface scratches.

Shape of the object

If the object it flat or has height variations can influence the choice of geometry. An object with large height variations might for example need a specific geometry to minimize occlusion. For more information about occlusion see *D07 – I have problems with missing data. What should I do?*

Resolution and speed requirements

Requirements on resolution and speed are often a big factor when choosing system geometry. If the application needs to run very fast a smaller angle might be needed. If on the other hand resolution is more important a larger angle might be the best choice.

See FAQ *L05 – How is the height resolution affected by the physical setup of the camera and laser* for more information.

What is going to be measured?

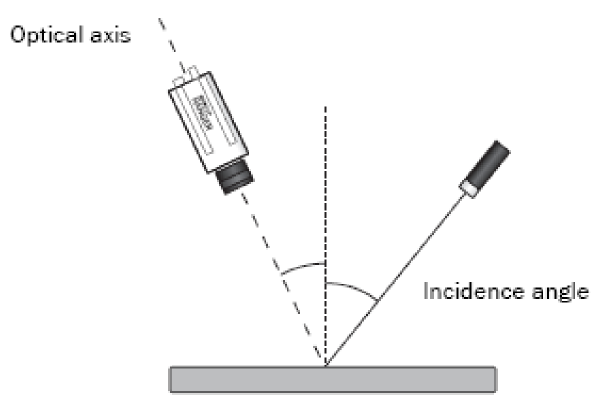
Different geometries will be needed depending on if we are doing range measurements, grayscale measurement or scatter measurement etc. For example during range measurement we need to have an angel between the laser and the camera to get any 3D data but for scatter measurements this angle needs to be minimized.

Range measurements

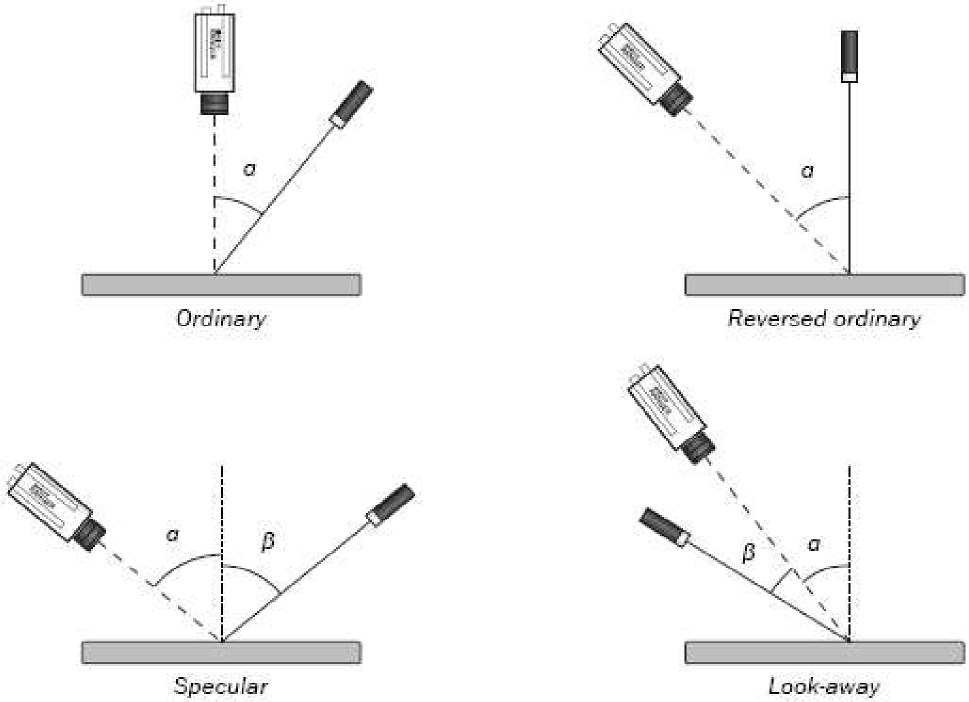
When measuring range, there are two angles that are interesting:

- The angle at which the Ranger is mounted
- The angle of the incoming light (incidence)

Both angles are measured from the normal of the transport direction. The angle of the Ranger is measured to the optical axis of the Ranger – that is, the axis through the center of the lens. In order to get correct measurement results, it is always important that the laser line is aligned properly with the sensor rows in the Ranger.



There are four fundamental geometries to consider: Ordinary, Reversed Ordinary, Specular, and Look-away geometry, see image below for definitions. Each geometry has its own characteristics and is applicable at a specific measuring situation.



Ordinary geometry

The ordinary geometry can give he highest range resolution. This geometry is good for Multiscan setups where the camera for example also can be used for 2D images with minimal geometric distortion. Diffuse reflex is used and therefore this geometry needs more laser power than the specular case. This geometry is not too sensitive for occlusion, i.e. the object itself is an obstacle for the laser line to illuminate all parts of the object, but the effect increases with an increasing laser angle. Moreover, on the front side of shiny objects, the mirror criteria might be fulfilled which can generate stray reflexes and thus result in corrupt height data. A drawback of this geometry is that the Y-position of height measurements is height dependent. Thus, for an object with height variations, the height measurements will be unevenly spaced along the Y-axis. This is also referred to as miss-register.

Reversed ordinary geometry

The great advantage with this geometry is that height measurements are taken from the same Y-position and thus, the height samples will be evenly spaced (assuming a constant movement of the object). This is the only geometry without miss-register. For issues like occlusion, light condition, and stray reflexions the conditions are similar as for the Ordinary geometry. However, an important difference is that given the same angle between laser and camera, the height resolution is lower than for the Ordinary case:

Specular geometry

This geometry is applicable for situations where the objects are dark or matte or when a low laser output power is required. The specular geometry is good for 2D gloss measurement and crack detection on shiny surfaces. This setup gives large variations on detected laser intensity depending on if the mirror condition is fulfilled or not. The geometry is bad for 3D measurements on shiny objects. Furthermore, it is more sensitive to occlusion than the above geometries and for large angles of α and β the miss-register will be heavy.

Look-away geometry

This geometry uses the diffuse reflex, but is more light consuming than Ordinary geometry since both camera and laser are directed away from the scene. The setup will be less affected by stray reflexes but the occlusion problem is more noticeable than for the other cases.

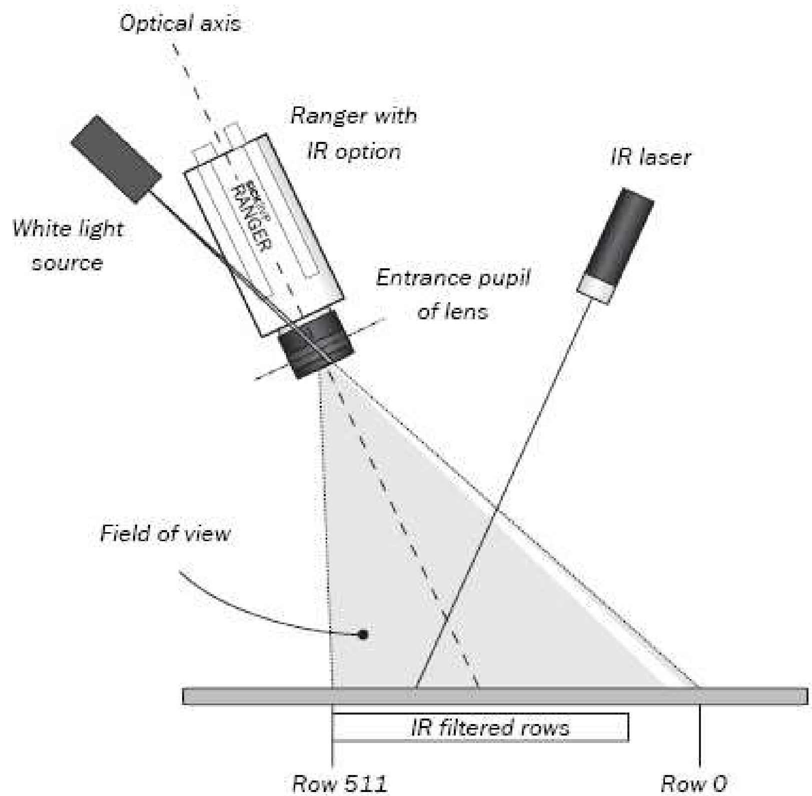
Intensity and Scatter measurements

For other types of measurements than range, a general recommendation is to align the light with the Ranger’s optical axis, or mount the lighting so that the light intersects the optical axis at the lens’ entrance pupil. By doing so, the light will always be registered by the same rows on the sensor, regardless of the height of the object, and triangulation effects can be avoided.

An exception is of course if gloss is going to be measured, since this type of measurement requires a specular geometry and usually a large angle. However, the triangulation effect is heavy if the objects vary in height. Therefore it is difficult – if not impossible – to measure gloss on objects that has large height variations.

When measuring with MultiScan, it is usually important to separate the light sources, so that the light used for illuminating one part of the sensor does not disturb the measurements made on other parts of the sensor.

If separating the light sources is difficult, the measurements may be improved by only measuring light with specific wavelengths, using filters and colored (or IR) lightings. For example, an IR band pass filter can be mounted so that it covers a part of the sensor, and an IR laser can be used for illuminating the object in that part. This way, range can be measured in IR filtered part of the sensor, and at the same time intensity can be measured in the non-filtered area using white light, without disturbing the range measurements. For certain Ranger models, a built-in IR filter is available as an option. The IR filter is mounted so that rows 0–10 are unaffected by the filter, and rows 100–511 are filtered.



For more information on how to setup a MultiScan setup, please see related FAQ's 005 and 017.

L02 – How do I choose a lens?

Selecting a lens for a straight-forward application is a three-step procedure:

1. Measure the object dimensions, or the maximum area in which the object can be located.
2. Measure the object distance.
3. Calculate the needed focal length.

In high-accuracy applications or under other special circumstances, selecting an appropriate lens may require special knowledge and considerations. The section about telecentric lenses below describes such an example.

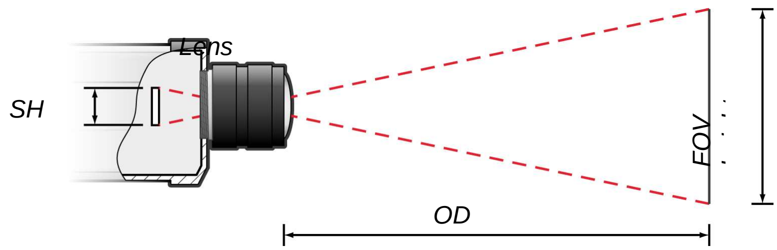
Calculation of Focal Length

A suitable focal length can be calculated with the following formula:

$$FocalLength = \frac{SensorHeight \cdot OD}{FOVheight},$$

where OD is the object distance.

The formula is a simplified model intended for practical use, not for exact calculations.



The figure above can just as well represent the sensor and FOV widths. The important thing is to be consistent in the formula by using only heights or only widths.

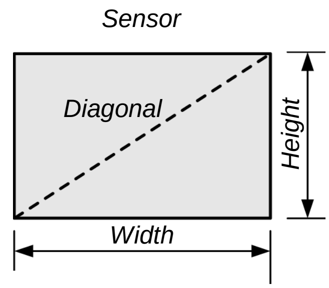
The sensor size is often measured diagonally in inches. SensorHeight in the formula above refers to the vertical height, which thus needs to be calculated from the diagonal.

Example

The focal length for a certain application needs to be calculated.

Known information:

- Camera: IVC-2D VGA (640x480)
- Sensor size 1/3 inch → diagonal 8.5 mm
- Object distance = 500 mm
- FOV height = 100 mm.



To use the above formula, the sensor height needs to be calculated first. The needed calculation principles are the Pythagorean Theorem to calculate the sensor diagonal in pixels, and the theorem about uniform triangles to find the sensor height in mm.

Calculations:

1. $SensorDiagonal = \sqrt{640^2 + 480^2} = 800 \quad pix$
2. $\frac{8.5 \ mm}{800 \ pix} = \frac{SensorHeight}{480 \ pix} \rightarrow SensorHeight = 5.1 \ mm$
3. $FocalLength = \frac{5.1 \cdot 500}{100} \approx 25 \ mm$

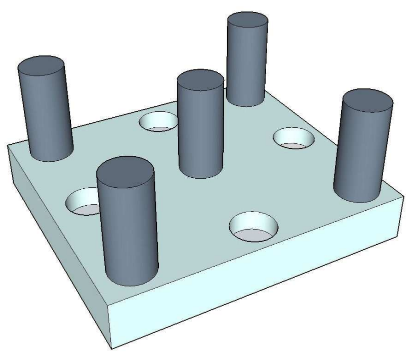
Telecentric Lens

Normal lenses give a slightly distorted image because of the view angle. **Telecentric lenses** can be used to reduce or eliminate such effects. The optics in a telecentric lens causes all light beams enter the lens in parallel. This has two practical consequences:

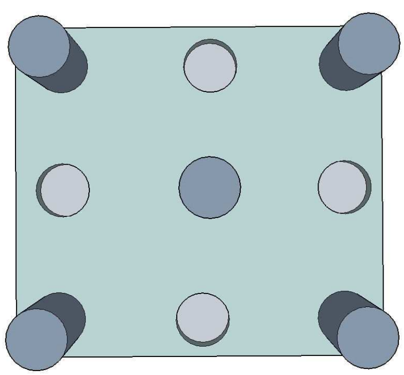
1. The lens diameter needs to be at least as large as the object's size.
2. The depth of field is deeper than for standard lenses.

Telecentric lenses are mostly used in high-accuracy measurement applications.

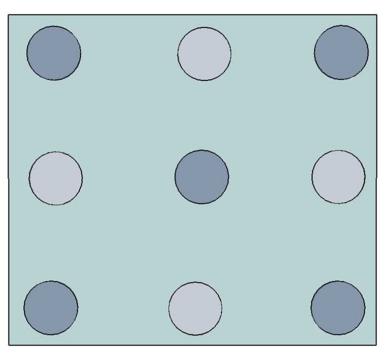
Example



Object with cylinders and holes. The task is to measure their diameters.



Cylinders as viewed through a normal lens from above.



Cylinders as viewed through a telecentric lens from above.

L03 – How do I choose laser?

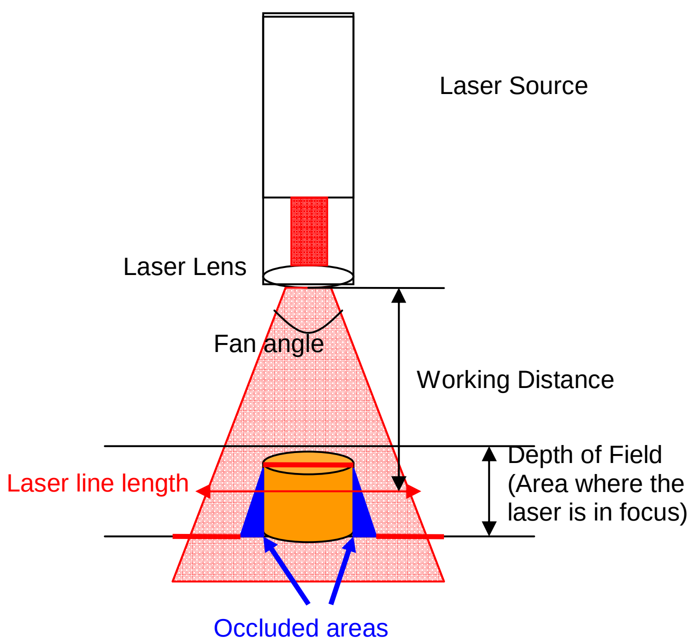
There are a number of things to think about when selecting which laser line source to use. Here is a list of things to consider:

1. Field of View
2. Laser occlusion
3. Resolution requirements
4. Required Exposure
5. Laser Safety
6. Lifetime

Field of View and Occlusion

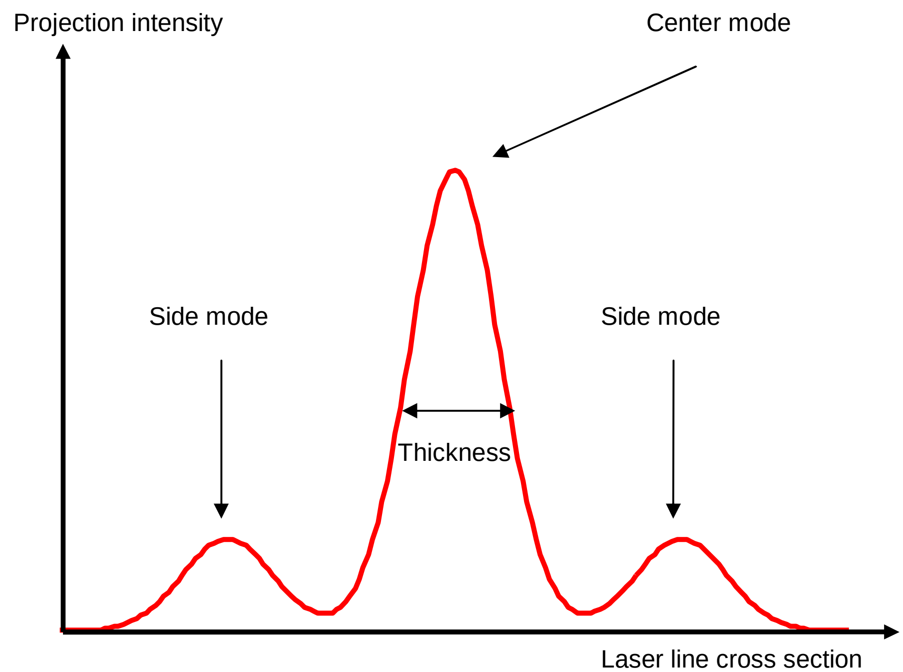
A laser line can be generated in several different ways. The most commonly used by Ranger and Ruler users is a point laser with an anamorphic lens. This lens stretches the projected laser point to a projection line. The properties of such a system are quite similar to those of normal camera lenses. There is a 'focal length', usually referred to as a *fan angle* and the depth of field, *working distance,* is limited.

Using a large fan angle the laser line becomes long also when the laser source is positioned fairly near to the object. The amount of laser occlusion increases. That is, there are larger areas next to objects where no laser light is projected. You should select a laser with a fan angle which gives you one as short laser line as possible but which covers the entire field of view



Resolution, Depth of Field and Thickness

The resolution of the measurement will depend on the laser line thickness. One easily makes the mistake of thinking that one as thin line as possible is always the best solution. This is not quite the case however. The thickness of the line can be used to trade resolution in Z (height) for resolution in X and Y (in the conveyor plane). A thin line can show very small details which gives a good resolution in X and Y. It is however more difficult to get a good measure of height since locating the laser peak is more difficult when the laser line occupies fewer pixels. To measure the height of a smooth surface, a relatively thick laser line is therefore better. If you wish to measure the height of very small components such as BGA:s or electrical connectors, the resolution in X and Y is more important and you need a thin line. Due to the optics of the laser, there is a limited depth of field of the laser line. That is, there is a minimum and a maximum distance from the laser between which the laser line is in focus. The farther away from these two boundaries the laser is projected, the thicker the laser line gets. You should also keep in mind when discussing the thickness of laser lines that the laser line is indeed not a line with a certain width but has a more or less Gaussian profile. Theoretically it is impossible to say how wide a Gaussian profile is. You just have to define how far below the peak value the laser line is to be considered to end. Beside the main Gaussian mode there may also be side modes (transverse modes). For most applications these are of no significance since they are sufficiently weaker than the center mode. If you are scanning shiny objects with a lot of sharp edges and corners, like electrical connectors, this may be a problem since specular (direct) reflections from a side mode may be brighter than the center mode of a diffuse reflection. For such applications it might therefore be important to really look into the finer details of the specifications of the laser projector.



There are basically three types of laser projectors with anamorphic optics.

- infinity focused lasers
- lasers with variable focus
- micro focus lasers

Infinity focus lasers are simply focused on infinity. There is a minimum working distance beyond which the laser can be said to be in focus. The laser line will of course become thicker the farther away from the laser you get but this will be in proportion to the increase in laser line length.

Lasers with variable focus can be modified to focus on different distances. The line is then focused at this distance and within a certain range from this distance. The farther from the working distance you go, the thicker the line gets.

Micro focus lasers are usually focused on a fix distance giving a fix laser length. The laser line is extremely thin at this particular distance but go thicker rapidly as you move away from the working distance.

Required Exposnor

In order to measure the position of the laser line you need sufficient exposure of the object. How much of the laser light hits the sensor is determined by the exposure time, the output power of the laser and the reflectivity of the object you are scanning. If you need to scan at twice the speed you need twice the amount of light. Remember that lasers are monochromatic light sources. That is, the light from a laser only contains light waves of one particular frequency. If your laser is red, red objects will appear as bright whereas blue objects will appear as dark in the image. Dark objects will of course also appear darker than bright ones. The only good way to know how much exposure you need is to try all the extremes of your application. Find the darkest and the brightest variants of your objects and see how their properties affect the acquisition results.

Laser Safety

Laser light can cause injury to both eyes and skin if misused. All laser devices are therefore classified into different classes. The higher the number the more dangerous the laser. Class I lasers can be used without any particular caution taken. Class II lasers may be dangerous if you look straight into the source. Class III lasers can be dangerous if you look at strong reflections of the light. Etc. If you feel uncertain about the safety issues, contact your local vision specialist or SICK IVP for advice. The choice of laser power may be limited by the environment in which the device is to be installed. There may also need to be safety equipment installed to ensure that the lasers can be disarmed during maintenance of the vision system.

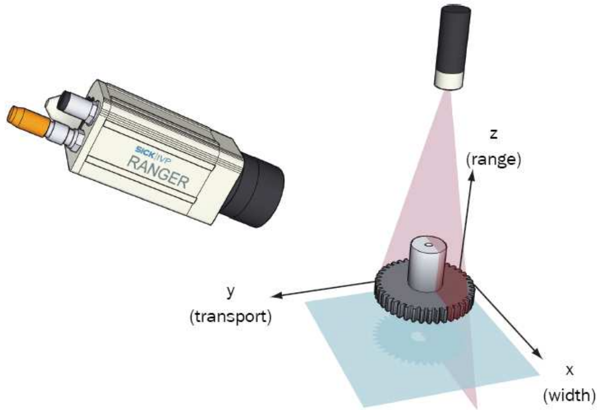
Lifetime

Lasers have limited lifetime. The lifetime is more or less defined by the time they are turned on. A rule of thumb is that the more powerful the laser the shorter it lives. Lifetime may be an important issue since the laser may be installed into a calibrated cabinet and quite difficult to replace.

L04 – What is the resolution of the data in X, Y and Z?

Coordinate system

To define the resolution of the data in X, Y and Z we first need to define the coordinate system that we are working with. The x axis is orthogonal to the conveyor movement, the y axis is in the transport direction and the z axis is orthogonal to the xy-plane. In the most common geometry, seen in the image below, the z axis is along the laser's optical axis.



X – resolution: The X resolution is dependent on the width FOV and the number of pixels on the sensor and can be calculated like this:

dX [mm/pixels] = width FOV / number of pixels

The width FOV can easily be measured by looking at a Ruler in image mode in Ranger Studio. The number of pixels in x for the Ranger camera is 1536 (except for Ranger D40/E40 which has 512 pixels in width). There is however one exception, for the RangerC55 and RangerE55 cameras there is one single high resolution sensor row with 3072 pixels that can be used for the HiResGray component.

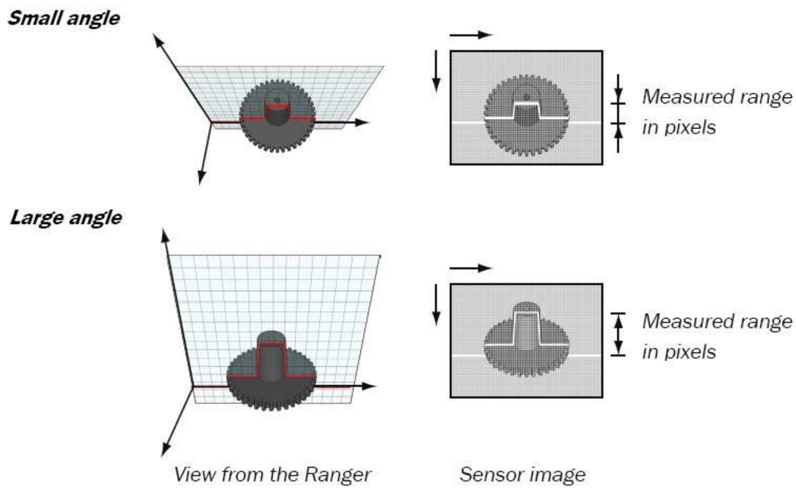
Y – resolution: The Y resolution can be seen as the profile distance. The Y-resolution has nothing to do with the X-resolution, nor the Z-resolution.
In free running mode the Y resolution is dependent on the object movement (conveyor) speed and the cycle time of the camera. The resolution in the Y-direction can be calculated as:

*dY [mm/scan] = Object movement speed [mm/s] * cycle time [s/scan]*

When using an encoder the Y resolution is calculated from the number of pulses per scan and the encoder resolution as well as the possible down scale of the trig signals to the camera.

*dY[mm/scan] = Pulses per scan / (encoder resolution * scale factor of trig signals)*

Z – resolution: The height resolution in a Ranger application is depending on the laser-camera setup, also referred to as the geometry of the measuring system.
A rule of thumb is that the height resolution increases with the angle between the Ranger and the laser. If the angle is very small, the location of the laser line will not vary much in the sensor image even if the object varies a lot in height. On the other hand if the angle is large, even a small variation in height would be enough to move the laser line some pixels up or down in the sensor image.



See FAQ with ID ***L05*** – *How is the height resolution affected by the physical setup of camera and laser* for approximations of the Z resolution for different geometries and a more detailed description of the Z-resolution calculation.

Note that these approximations give the resolution for whole pixels. If the measurement is made with sub-pixel resolution, the resolution in the measurement is the approximated resolution divided by the sub-pixel factor. For example, if the measurement is made with the Hi3D component that has a theoretical resolution of $1/16_{th}$ pixel, the approximate resolution is ΔZ/16.

Normally one uses the sub-pixel resolution approximation of $1/10\text{-}1/12^{th}$ for the Hi3D component.

L05 – How is the height resolution affected by the physical setup of camera and laser?

The height resolution in a Ranger application is depending on the laser-camera setup, also referred to as the geometry of the measuring system. There are four fundamental geometries to consider: Ordinary, Reversed Ordinary, Specular, and Look-away geometry, see Figure 1 below for a definition. Each geometry has its own characteristics and is applicable at a specific measuring situation. A rule of thumb for all four geometries is that an increased angle between the camera and laser will result in an improved height resolution.

The height resolution can be reasonably well approximated in terms of the width resolution (the spatial resolution across the sensor). The approximation is at least true for reasonably ordinary angles and distances between laser and camera. In the following, it is assumed that the width resolution is known or can be approximated. A sufficiently good approximation in this context will be to divide the maximal visible area (i.e. the field of view at a specific height position) by the crosswise resolution of the sensor. The following notation is used in the four different expressions for the height resolution (see Figure 1 for a definition of the coordinate system):

- ΔZ = height resolution (length unit / pixel)
- ΔX = width resolution (length unit / pixel)

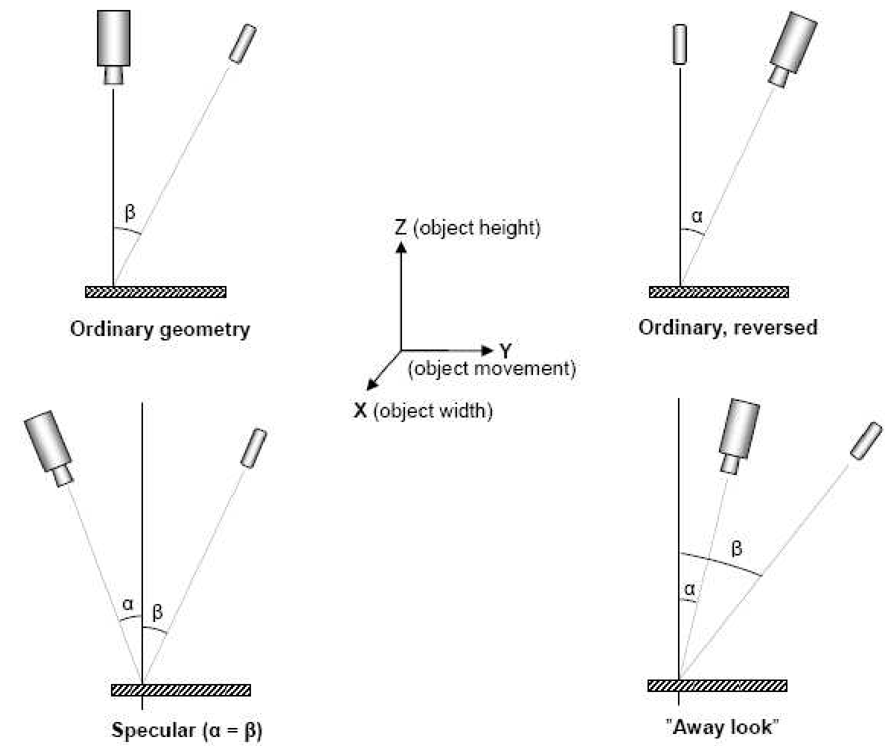


Figure 1, Geometry defintions

Height resolution defined

The height resolution is a description of the height increment needed (in Z-direction) to move the projected laser line exactly one row on the sensor. In a plane parallel to the sensor where the width resolution is ΔX, the resolution along the perpendicular axis (S, in Figure 2) will also be ΔX. This, since the sensor pixels are square and equally spaced in both directions. The sought increment, ΔZ, in the Z-direction (i.e. the height resolution) that causes a movement of ΔX along the S-direction is determined by the intersection of the laser plane and a straight line from the camera to ΔX along the S-axis (see figure 2). The height resolution is identical to the Z-position of that intersection.

To simplify the expressions, a fair approximation of the height resolution can be made by finding the intersection of a line from ΔX along the perpendicular *C-direction* onto the laser plane instead of finding the intersection of a straight line from ΔX to the camera, see figure 2b. ΔZ is then derived as above by finding the intersection's Z-coordinate. With this approximation the angle difference is neglected, but for most situations the angle difference will be small and thus, the approximation will be sufficiently accurate.

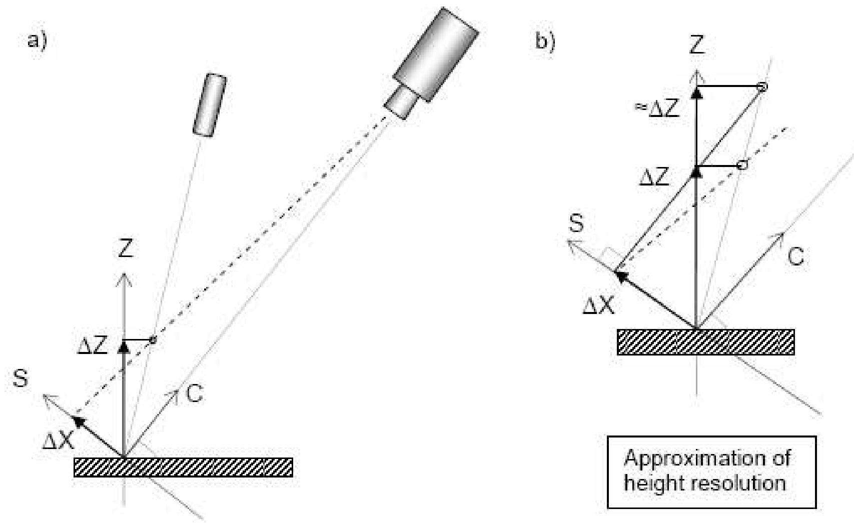


Figure 2, Definition of Height resolution, ΔZ, and its approximation

Note that the below expressions concerns the height resolution without sub pixel resolution. To get the correct expression for sub pixel resolution, the below approximations should be divided with the sub pixel factor (e.g. 2, 4 or 10).

Ordinary geometry: $\Delta Z \approx \Delta X / \tan(\beta)$

Reversed ordinary geometry: $\Delta Z \approx \Delta X / \sin(\alpha)$

Specular geometry: $\Delta Z \approx \Delta X * \cos(\beta) / \sin(\alpha+\beta)$
For the specular case where $\alpha=\beta$ the formula simplifies to: $\Delta Z \approx \Delta X / 2*\sin(\alpha)$

Look-away geometry: $\Delta Z \approx \Delta X * \cos(\beta) / \sin(|\alpha - \beta|)$
Note that an absolute value is taken around $\alpha - \beta$, this makes the formula valid also if camera angle is greater than laser angle.

Configuration and Parameter Optimization

O01 – How can I optimize the cycle time?

The scan rate and the camera integration time are inversely proportional; maximizing the scan rate is equivalent to minimizing the integration time. There are mainly two parameters that affect the integration time:

- The height of the measurement ROI (Region Of Interest, number of sensor rows used to extract the range data)
- The choice of 3D algorithm and its parameters

Reducing the measurement ROI height

The first thing to do when trying to minimize the integration time is usually to reduce the number of rows used on the sensor to as few as possible. How much you can reduce the number of rows of course depends on how high your object is and what geometry you use (Ranger, else fix geometry).

If you are working with Ranger Studio switch to image mode configuration and make sure that the Measurement ROI overlay parameter set to 1 (one). Your measurement ROI will then be drawn on the sensor image as white dotted lines so that you can easily see which sensor rows are being used. Change the Start Row and Number of Rows parameters so that the ROI barely covers your object (perhaps you will need some extra rows to compensate for conveyor vibrations or object height variations). Move your object back and forth to make sure that the laser line always stays completely inside the ROI.

Selecting an appropriate 3D algorithm

If reducing the height was not enough you need to change 3D algorithm or alter the parameters of the selected algorithm. Remember that changing the measurement method to a faster 3D algorithm will most likely have a negative impact on the resolution of your measurements. (This does not apply to the HorMax algorithm which both gives a lower resolution and is slower than the HorThr algorithm).

	Scan Rate [pfs / s]	**Pixel Resolution**
Hi3D	500 – 4,250	1 /10
HorMax	500 – 8,000	1 /2 – 1
HorThr	5,000 – 10,000	1 /4 – 1 /2

If you can live with the lower resolution of the HorThr algorithm compared to the Hi3D this is usually the way to go. Not only gives the HorThr the fastest possible scan rate using a smaller ROI, but it also retains a high scan speed when the height of the measurement ROI increases.

If you need to stick with the Hi3D or HorMax algorithm there are a few things you can change in order to speed up the algorithms a bit:

- Reduce the *Ad Bits* parameter
- Set the *Binning* parameter to 2 (two)

Refer to the charts in the Measurement Components section in the manual to see how these changes affect the scan speed.

Use Implicit Reset Instead of Explicit Reset

- Pro: Faster
- Con: Conveyor speed-dependent exposure time if externally triggered

Implicit reset is faster than explicit reset since it doesn’t use an extra reset ramp. To use implicit reset set the exposure time parameter to something equal or greater than the cycle time. The downside of using implicit reset comes when running in triggered mode where the exposure time varies with the speed of the conveyor. In this case you may want to consider using *Mark* to measure the inter profile distance instead of using profile triggering or flashing the laser.

Finishing up

Now there is just one last thing to do; set the *Cycle Time* parameter to as little as possible. If you are using Ranger Studio, switch to measurement mode. Now *increase* the cycle time parameter slightly and you should be presented a message in the log window telling you something about the *approximate minimum cycle time*. Now change the cycle time parameter to this minimum value (you will probably need to add a few us since it's just an approximate figure).

I did all the above, but now I get a lot of missing data!

When running at high speeds and measuring dark objects or if your conveyor is made of a dark material it is common to get some missing data. At this point you could:

- (Read the FAQ with ID D07 that handles the missing data problem.)
- Increase the cycle/exposure time as much as possible without jeopardizing the speed requirements of your application.
- Try to increase the *Gain* parameter
- Open up the aperture (will have a negative effect on the depth of field)
- Use a stronger laser (3B option for the Ruler).
- Use an additional laser aligned with the first
- Use a more specular geometry
- Change the colour of the conveyor

O02 – How can I increase the scan rate of my camera?

See answer in O01 above.

O06 – What is the difference between cycle time & exposure time?

Generally speaking the *exposure time* is an individual parameter of a component and is the time during which the corresponding sensor pixels are exposed. The *cycle time* is the global time between the acquisition start of two consecutive profiles. The cycle time is either determined by the camera internal clock (constant) or by encoder trig rate (conveyor speed dependent) depending on the chosen trig mode.

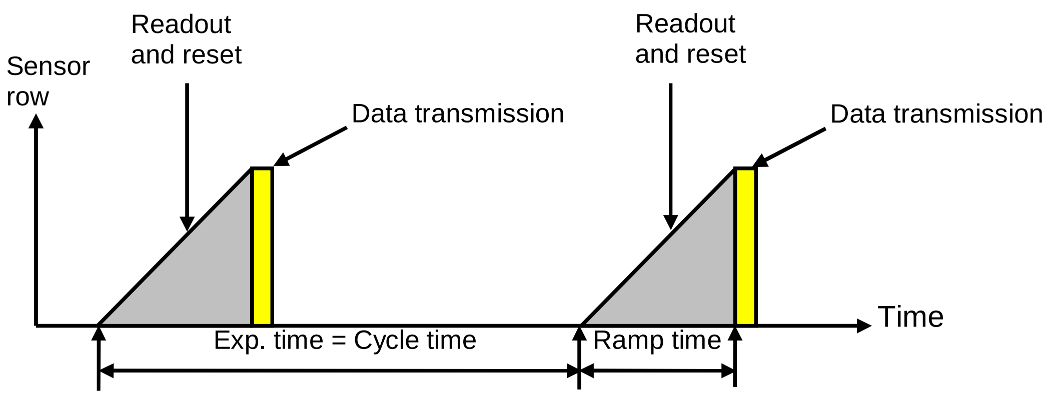
The M12 CMOS sensor found in both RangerM/C/D/E & RulerE cameras has a rolling shutter, which means that reset and readout is done row by row. Each reset and readout takes a certain amount of time illustrated by the ramps in the figures below (the ramp time is algorithm dependent, typically 0.5 – 10 µs/row for 3D algorithms).

Depending on how you configure your camera you can have the readout of the last profile work as the reset for the next (*Implicit Reset*) or have an extra reset ramp before the readout ramp (*Explicit Reset).* In order to have a constant exposure time over each measurement area the inclination of the reset ramp must be the same as that of the readout ramp.

It is the relationship between the cycle time and exposure time settings that determines if *Implicit* or *Explicit* reset is to be used by the camera.

- Exposure time ≥ Cycle time → *Implicit Reset*
- Exposure time < Cycle time → *Explicit Reset*

Implicit Reset – (Exp. Time ≥ Cycle time)



Example of simultaneous reset and readout of a single 3D component (Implicit Reset). The readout ramp of the previous profile works as a reset for the next profile.

The readout of the last profile works as a reset for the next. The exposure time parameter is neglected, the time the sensor pixels are exposed is equal to the cycle time. In triggered mode this means that we can get a non constant exposure time.

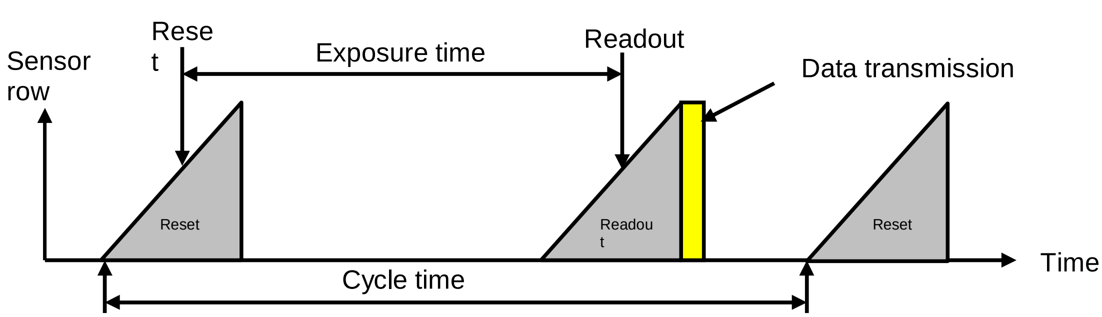
Pro: Allows for fast profile rates

Con: Conveyor speed dependent exposure (if externally triggered)

Speed: Min cycle time ≈ ramp time

Implicit reset is recommended for non multiscan applications where a high speed is crucial and the conveyor speed is constant.

Explicit Reset – (Exp. Time < Cycle time)



Example of sensor reset and readout of a single 3D component (Explicit Reset). An extra reset ramp allows for a constant exposure time independent of conveyor speed.

Explicit reset is recommended whenever the conveyor speed is variable and external triggering is used. An extra reset ramp is used before the readout ramp to ensure a constant exposure time.

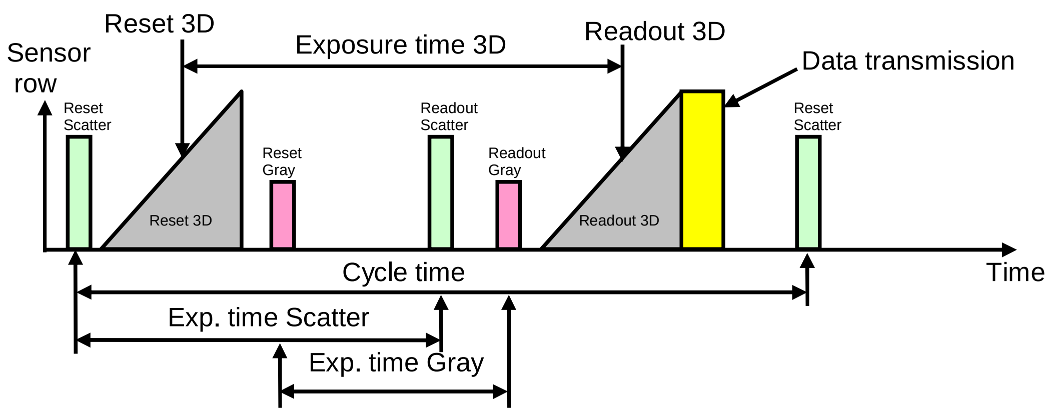
In triggered mode the cycle time parameter is neglected and the cycle time depends only on conveyor speed. In free running mode the *cycle time* is internally controlled and constant.

Pro: Well-controlled, constant exposure time, independent on conveyor speed if externally triggered

Con: Slower max profile rate

Speed: Min cycle time ≈ 2*ramp time

Explicit reset is also suitable for multiscan setups where the *exposure time* of the different components often need to be shorter than the *cycle time.*



Example of a multiscan configuration with three components (3D, Gray & Scatter), each with an individual exposure time (Explicit Reset).

O08 – Which 3D algorithm should I choose?

The following 3D algorithms are available for the 3D cameras

- Horizontal threshold (HorThr)
- Horizontal maximum (HorMax)
- Horizontal maximum and threshold (HorMaxThr)
- Hi-resolution 3D (Hi3D)

Note that not all algorithms are available for all cameras. See the reference manual for more information.

Before choosing which 3D algorithm to use in your project you should always write down the requirements in terms of speed & resolution of your application. If your application requires high scan speed you probably want to go for the HorThr algorithm. If you need high z-resolution you should probably go for the Hi3D algorithm.

Which algorithm to choose can also depend on the characteristics of your object. If you are having problems with false reflections, trying a different algorithm can sometimes help. For instance the HorThr algorithm can be parameterized to search for the laser line from top to bottom or vice verse. This can be useful if you have a reflection always appearing on the same side of the true laser line. If you have problems with false reflections on both sides of the true laser line the HorMax algorithm could yield better results as it derives the laser line position based on the sensor row with the greatest intensity value.

Here follow a short summary of the characteristics of each algorithm

Horizontal threshold (HorThr)

This component is used when the most important requirement is high speed. If a smaller sensor region is used up to 35 000 profiles can be acquired per second. The range resolution depends on the number of thresholds used. Using one threshold gives ½-pixel resolution, while two thresholds gives ¼-pixel resolution. Using two thresholds will increase the execution time of the component, hence decreasing the maximum speed performance.

Horizontal maximum (HorMax)

The HorMax algorithm can be useful when the image contains several reflexes, but only the strongest reflection is of interest. The range resolution is less accurate compared to HorThr, but the component delivers an intensity profile as well as a range profile in each scan. The intensity measurement can be used to e.g. determine gray scale properties of the object along the laser line.

Horizontal maximum and threshold (HorMaxThr)

The HorMaxThr component combines the HorThr and HorMax components. It uses the HorMax algorithm for intensity data and the HorThr algorithm for range data. This component gives better range measurements than HorMax, and is also slightly faster. One threshold is used, resulting in ½ pixel resolution in range measurements.

Hi-resolution 3D (Hi3D)

The Hi3D component is the component to use when high resolution is the most important requirement and speed demands can be renounced. This component measures range with a resolution of 1/16th pixel, which is the best resolution of the available range components. In addition to range, the Hi3D component also measures intensity.

There is also a variant of the Hi3D component called Hi3D COG. This component performs the measurement using an algorithm called Center of Gravity and delivers the data in a somewhat different way. See the reference manual for more information.

O09 – What is mark data & when should I use it?

Mark data can be used if you want to find the actual distance between two profiles even if the Ranger/Ruler is measuring in free-running mode, as long as you have an encoder connected to the camera. The encoder information can then be embedded in the profiles sent to the PC as *mark data*. Your application can then use this information to calculate the distance between the profiles.

The mark data is supplied as a separate sub-component in each component in the data format. Refer to the reference manual if you need a more detailed description on how to access the mark data.

If your application requires high speed this could be a good setup as you can run the Ranger/Ruler in implicit reset mode without having to worry about conveyor speed dependant exposure times. You can then use the mark data to calculate the inter profile distance. Depending on your application this might demand a higher skilled programmer as the sampling distance in *y* (movement direction of the conveyor belt) varies with the conveyor speed.

O10 – Why do I get error messages when changing parameter values?

Many of the parameters are related to each other, meaning that the valid range of one parameter can depend on the setting of some other parameters. When a parameter is changed the camera tries to optimize the performance (find the minimum cycle time) based on the current settings. If some parameter setting is invalid you only get the feedback that there was an error in setting this combination of parameters. What the exact problem is is not stated. In this case you need to figure out what effect the parameter setting you just did could have on other parameters and thereafter change the parameter again in a way that gives an overall valid setting, or change the related parameters such that you can make the desired setting.

Typical parameters that affect each other are, sorted by algorithm:

Hi3D: Cycle time, exposure time, number of rows, A/D bits

Horizontal Threshold: Cycle time, exposure time, number of rows, number of thresholds

Horizontal Maximum: Cycle time, exposure time, number of rows

Scatter: Exposure time direct, exposure time scatter, cycle time

See also FAQ question O16 for related information. FAQ questions O01 and O06-O08 is recommended if you want to learn more about the parameter settings for different algorithms and how cycle time and exposure time are related.

011 – What is the difference between image & measurement mode? Do I need both?

The Ranger/Ruler camera has two different modes (or configurations): *Image* & *Measurement.*

In the image configuration the camera acts as a 2D camera and delivers gray scale images to the PC (raw sensor images). Most users only use this configuration as assistance when setting up the ROI, exposure time & gain parameters of their measurement components.

The measurement configuration is what most users will be using when their application is up and running.

The measurement data is delivered to the PC as scans, where each scan contains the results from measurements made at one point in time (think of the Ranger as a line scan camera capable of performing several measurements at the same time). Due to the MultiScan functionality of the Ranger, each scan can contain one or more profiles with different measurement data (ex. range, scatter, gray).

O12 – How does Ruler scatter work and what is “Scatter Offset”?

Ruler Scatter is, in contrary to Ranger Scatter, measured using the range laser and is always dual sided. Dual sided meaning that the camera calculates the scatter as the mean scatter value from both sides of the laser line.

The *Scatter Offset* parameter corresponds to a percentage of a “maximum position” which is eight pixels away from the centre of the laser line. The reason to use this relative measure is to reduce effects of the distance to the sensor. Basically, laser lines reflected on surfaces near to the sensor will appear wider than those reflected on surfaces farther away. The side band will therefore move farther away from the centre line when looking at objects at shorter distance to compensate for this. This feature is not available in the Ranger since the scatter line in that case does not contain any height information.

O13 – Can the Ranger/Ruler normalize the scatter data?

Neither the Ruler nor the Ranger can normalize your scatter data. If you find it necessary to normalize the data you will have to do so yourself. One way of doing this could be to divide the scatter data with the intensity data. In order to get the effect of a longer exposure time the parameter “gain scatter” can be used.

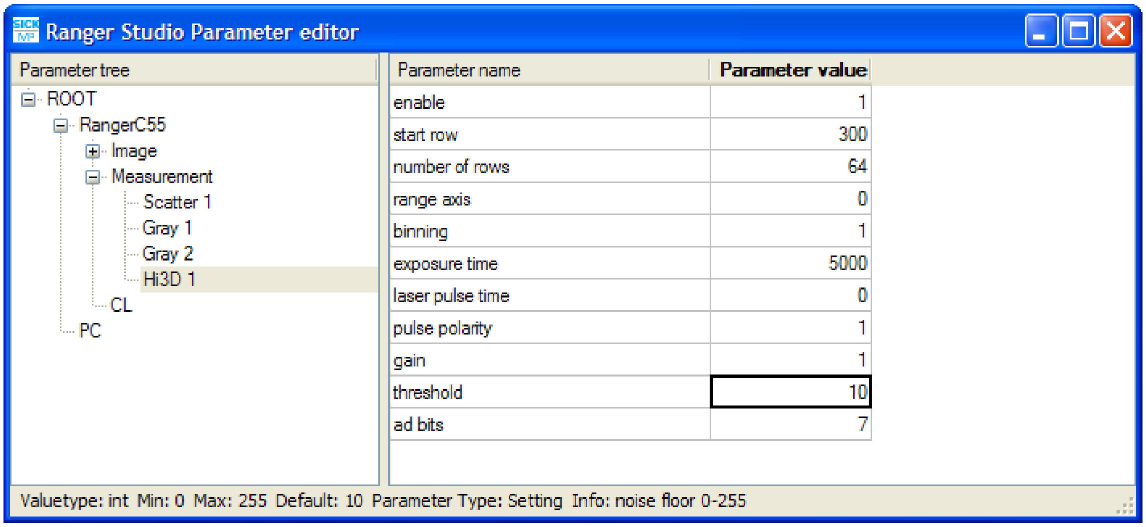
O16 – How do I configure the parameters of a certain component? How do I know the valid parameter value range?

Setting Parameters

In Ranger Studio there is a parameter editor window in which you can configure the parameters to fit your needs. You find this by pressing Edit in the Parameters tab on the right in the main Ranger Studio window. If you expand the parameter tree you will find the parameters for the Image mode as well as for the Measurement mode. Under Measurement in the tree are the different components, i.e. different sensor algorithms (e.g. Hi3D, Scatter, Gray).

For each component there is a list of parameters that can be modified for that component. Click on the parameter you would like to edit. In the bottom list of the window you will see an explanation of that specific parameter and the allowed values for the parameter.

Some parameters are gray and cannot be edited while the camera is running. This is because they will either change the data format of the data sent to the PC (in this case the frame buffers need to be re-allocated before starting again) or the sensor program needs to be re-optimized due to the changes in parameter values. In either case you need to stop the camera (by clicking ‘Stop’ in the main window of Ranger Studio).



Valid Parameter Range

The valid range for parameters is not always easy to predict. The range given in the tooltip at the bottom of the window only states the theoretical limits for the parameter. In many cases the actual range will depend on how other parameters are set. The program running on the sensor chip is generated when the parameters are sent to the camera and are being optimized in the camera to allow for maximum performance. The only feedback from this optimization is if it was successful or not and the minimum cycle time that can be set given all the other parameters.

Typical parameters that affect each other are, sorted by algorithm:

Hi3D

Cycle time, exposure time, number of rows, A/D bits

Horizontal Threshold

Cycle time, exposure time, number of rows, number of thresholds

Horizontal Maximum

Cycle time, exposure time, number of rows

Scatter

Exposure time direct, exposure time scatter, cycle time

In MultiScan configurations optimisation of the sensor program is quite complex which makes it more difficult to predict the parameter ranges. See FAQ question O19 on how to configure a MultiScan configuration for details on this.

O20 – Can I find out the distance between two scans in free-running mode?

Yes you can! To measure the distance between two scans when not running in profile triggered mode you can configure the camera to embed encoder counter values in each profile (you will of course still need to have an encoder connected to the camera). This is done by setting the mark parameter to 1.

The mark data is supplied as a separate sub-component in each component in the data format. Please, refer to the reference manual for a more detailed description.

O21 – What is the optimum working temperature of the Ranger and how do I reduce thermal noise?

There are no specific figures for surrounding temperatures in which the camera will function best, but the recommendation for the Ranger E camera is that the ambient temperature is somewhere between 5-45℃.

For the Ranger C camera on the other hand the temperature values referred to regards the camera housing and should be somewhere between 5-50℃ during operation. As long as the temperature is within this span the camera should work well.

The storage temperature of all Ranger models range from -20 to 70 ℃ in a non condensing environment.

If you experience problems with noise in your images try to keep the temperature as low as possible as this will reduce the amount of “leakage currents” on the sensor, and thus give better image quality. The higher the temperature the more thermal noise will affect your images. In general, this phenomenon doubles approximately every ten degrees Celsius increase.

This issue becomes more evident in dim environments when dark and difficult materials are to be scanned (black rubber for example). What you can do to reduce this unwanted characteristic – other than lowering the temperature – is to try and optimize the geometry. For example, a specular view angle can be used when dark materials that do not reflect much light, is to be scanned. Normally, the small pixel deviations caused by a high temperature of the sensor are rarely an issue if you tweak the physical setup somewhat.

In order to maintain a low working temperature, the heat produced by the camera can be lead away by a metal plate attached to the device.

Programming environments and compatibility

P06 – Are my old Ranger C/M applications compatible with the Ranger E?

Ranger M

The ICON API used to program the Ranger E is not compatible with the Ranger M. The differences are quite large so you should best rewrite the entire image acquisition part of your application from scratch.

Ranger C

Ranger C uses ICON as well as the Ranger E. If your application is written using a fairly recent version of ICON you will only need to rewrite the initialization parts of your acquisition program. The older versions of ICON did not give any support to communicate with the frame grabber meaning that all your data shuffling is done through your frame grabber's API. You can leave this the way it is if you are already satisfied with the result. Nowadays it is otherwise possible to use ICON also to get the images into your software without using the frame grabber's API. This is so far only the case for Coreco grabbers, though.

External Triggering

T01 – How can I trigger the Ranger with external devices?

There are two ways in which external triggering can be used:

- to control when the camera start to scan a series of scans
- to tell the camera exactly when one scan should be acquired

The first point above is typically controlled using a photo switch. If this data acquisition is enabled (“Use Enable” is set to 1) the camera will only acquire scans when the signal from the photo switch is high. If the Use Enable parameter is disabled (set to 0) a continuous flow of data is sent to the PC.

The second point above can be controlled by having an encoder connected to the conveyor belt. The encoder sends pulses to the camera and you can tell the camera to acquire and send a scan on every n:th encoder pulse (“trig mode” parameter is set to 2 for pulse triggered mode and set to 0 for free running). This ensures that the distance between the profiles is equal (equidistant), resulting in constant y resolution and correct object proportions (if square pixels). This is useful if the conveyor belt is not moving with a constant speed. The opposite of encoder triggered scanning is to use free-running mode. This means that it is constant time between consecutive scans. This is useful when measurements are only critical in x and z, or if the conveyor movement is constant.

For electrical connections, please refer to the Ranger hardware manual and the Ranger E/D operating instructions from the installation of the Ranger SW.

T02 – How do I connect an encoder to trigger profile acquisition?

There is a difference in how to connect a RangerC camera compared to a RangerD/E or Ruler camera. Below is first RangerD/E connections explained and later RangerC in a separate chapter.

RangerE

The RangerD/E and Ruler products all have 5V differential encoder inputs A, A-, B and B-. All encoder inputs have pull-up resistors to 5V meaning we don't have to connect signals that should always be at 5V level.

In RangerD/E and RulerE products the IO inputs for encoder and MARK data information are the same physical input pins (not as in RangerC where the encoder input and MARK input are different IO pins).

First let's have a look at the pin connections for the RangerD/E and RulerE cameras

Ranger E/D encoder connector			SICK Stegmann encoder			Ranger E/D power-I/O terminal
Pin	Color*	Signal	Pin	Color	Signal	Screw terminal
1	White	In_A+	5	White	A+	3
2	Brown	In_B+	8	Pink	B+	4
3	Green	In_B-	1	Black	B-	7
4	Yellow	In_A-	6	Brown	A-	6
5	Gray	GND	10	Blue	GND	15

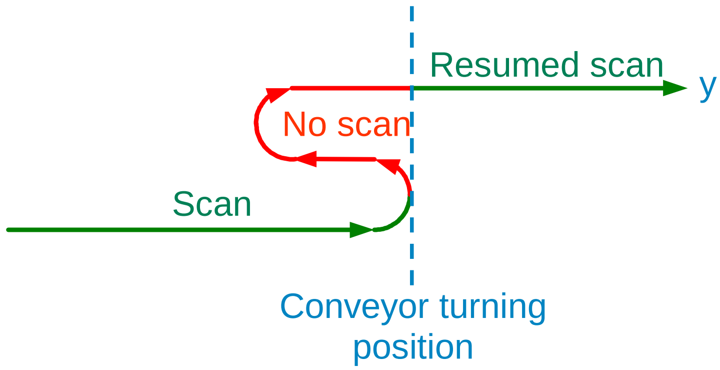
*Color is valid for cable type STL-1208-

You can choose to connect the encoder in two possible ways:

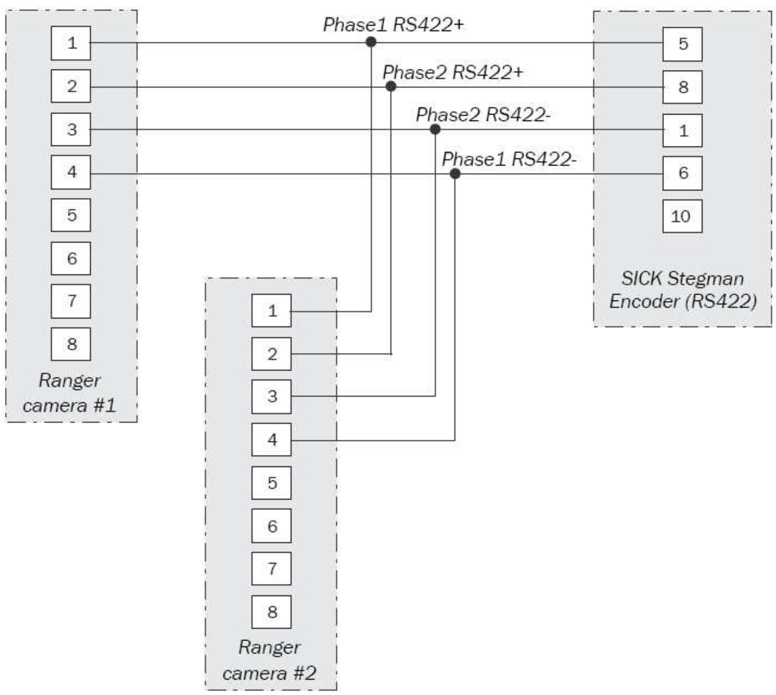
- Scanning in one direction (forward) only
- Scanning in both (forward and backward) directions

Scanning in one direction

Scanning in one direction only is used to avoid that the same object is scanned twice. When the conveyor starts to move backwards, the counter starts to count down. The scan is not resumed until the conveyor moves forward again and has reached the turning position. One way scanning requires two encoder channels (A/A- and B/B-) to control both the forward and the backward signal, and thereby keep track of the movement direction.



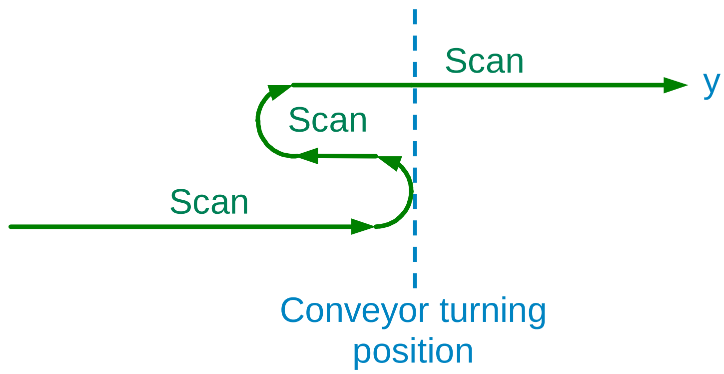
The following image shows a wiring diagram for connecting one Sick Stegmann RS422 encoder (DRS-60 with TTL output levels) to two Rangers to control both forward and backward signals.



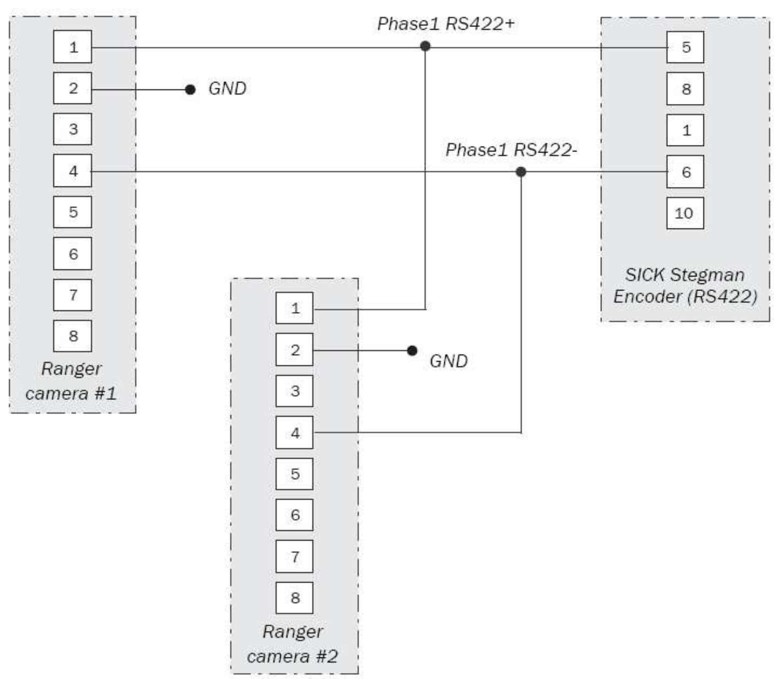
Figur 1. Wiring example – forward and backward signals

Scanning in both directions

When allowing scanning in both directions the camera does not care if the conveyor moves forwards or backwards. One encoder channel is enough (B/B-) and all pulses from the encoder will be interpreted as forward signals.



The following image shows a wiring diagram for connecting one Sick Stegmann RS422 encoder (DRS-60 with TTL output levels) to two Rangers, where all pulses from the encoder are interpreted as forward signals.



Figur 2. Wiring example – forward signals

Ranger C

The RangerC camera has non-differential 5V TTL inputs. There is **NO** pull-up for any input pin which means that all input pins must always be connected. In the RangerC camera there are two internal counters, one is used for encoder input and the other one is used for MARK data input. The encoder inputs are In2 and In1, and the Mark inputs are In3 and In4.

Connecting an encoder

As for the Ranger D/E cameras it is possible to wire the cables in such a way that you can scan in both directions (clockwise or counter clockwise rotation of the encoder) or just one forward direction (A-pulse from encoder is 90 degrees ahead of B-pulse). The table below describes how to connect the cables in order to trigger in one or both directions.

Trig in both directions		Trig in one direction	
Ranger C camera	Encoder	Ranger C camera	Encoder
In4	A	In4	A
In3	Gnd	In3	B

MARK Connections

Mark Specifies whether or not to include encoder values or scan id's, and input status information in the scan.

The MARK signal should be connected to the other internal counter, using In2 and In1 in the same way as for the encoder when using In4 and In3 instead.

The following table describes how to connect an external signal to the MARK data IO pins.

Encoder increments MARK data in both directions		Encoder increments MARK data for forward movement only (same functionality as for encoder input)	
Ranger C camera	Encoder	Ranger C camera	Encoder
In2	A	In2	A
In1	Gnd	In1	B

If no external signal is connected, then MARK data can still be included in the scan, but then only as profile ID counter since no encoder information is available.

T03 – Which encoders are supported?

The Sick Stegmann RS422 encoder (DRS-60or 61 with TTL output levels) is supported for the 3D cameras. This is an incremental encoder that can be used either for triggering the camera to make measurements or to mark the data with the encoder counter value.

For wiring instructions see appendix E in the manual for each product or FAQ T02 or T11.

In theory, any incremental encoder with 5V differential level of the output signals A, A-, B and B- should work, but we have not verified this ourselves.

T04 – How fast can I trigger the Ranger/Ruler?

How fast you can trigger the camera mainly depends on the cycle time (or exposure time if you are running in explicit reset mode). You can never trigger the camera faster than the minimum cycle time, if so is done you will get the following over trig error message:

```
   <RulerCam-err_proto> Error: 44011 : Trace: 9. The trig rate was too high, scans
lost.
```

Each time you change a parameter and have selected measurement mode you will get a log message which indicates the approximate minimum cycle time:

```
   <RulerCam-err_proto> Message: Approximate min cycle time is 427 micro
```

T05 - Is the Ranger/Ruler sensitive to noise on the encoder input?

The answer to this question depends on whether you are using a Ranger C or one of the Ethernet based cameras.

Ranger E / Ruler E

The encoder input is a differential input which makes it fairly robust to noise. If there is an electrical disturbance in the surrounding of the encoder cable, this disturbance is likely to induce a voltage in the cable. All the wires of the cable will be affected in the same direction though (positive or negative). The point with the differential input is that such outer disturbances do not affect the function of the encoder since the state of a certain input, e.g. A, is determined by looking at the relative difference between A+ and A- rather than at just one of the two. If A+ has been increased with e.g. 2 volts, A- will also be increased by 2 volts. The difference between the two will therefore be unaffected.

There are however some types of disturbances which the Ranger E and Ruler E are sensitive to. A normal cycle of the two signals A and B (which is what remains after decoding of the differential input) is: AB = [00, 10, 11, 01]. After one such cycle the encoder counter will count up one step. If the reverse cycle appears, AB = [00, 01, 11, 10], the encoder counter will count down one step. With the current encoder counter implementation the encoder counter will however count up one step already if a cycle AB = [00 10] appears. This means that if there are single pulses on channel A the counter will count upwards. This may be a problem in applications where a conveyor belt stops while scanning and where there are vibrations in the system. This problem will be solved in future versions of the camera FPGA.

The system may also be sensitive to glitches.

Ranger C

The Ranger C is more sensitive to noise on the encoder input since there is no differential input. The A and B signals are simple TTL level inputs which means that disturbances of the type described above may affect the counting. More than that the input is sensitive to the same kind of cycles as described above in the section about E cameras

T06 – I get the message 'trig rate was too high, scans were lost'. What should I do?

You are using external profile trigging, probably using an encoder. The message means that given the current configuration the camera is not able to scan at the rate demanded by the current conveyor speed and encoder settings. There are two ways of resolving this problem:

1. Reduce the resolution in Y (along the conveyor) by increasing the number of 'Pulses per profile' in the Measurement parameters. This will increase the distance between two adjacent profiles which means you will be able to scan at higher conveyor speeds.
2. Reduce the minimum cycle time by modifying your parameters. See the FAQ article about reducing cycle time for details. The top three things to try are:
    a. Reduce the number of rows used on the sensor. For 3D profiling
    b. Reduce the exposure time (if using explicit resets)
    c. If you are using Hi3D, try using Horizontal Thresholding instead

The problem could also appear because of noise on the encoder signal. Make sure the encoder is properly connected (See the FAQ articles or manual on how to connect an encoder). Investigate if there could be noise or vibrations in the system.

T11 – How do I make the encoder count in both or only one direction?

See answer in T2.

T12 – What are the relationships in between: enable signal - scan height - number of scans per frame – free running or encoder controlled acquisition?

Camera setup example

Moving camera system and still object

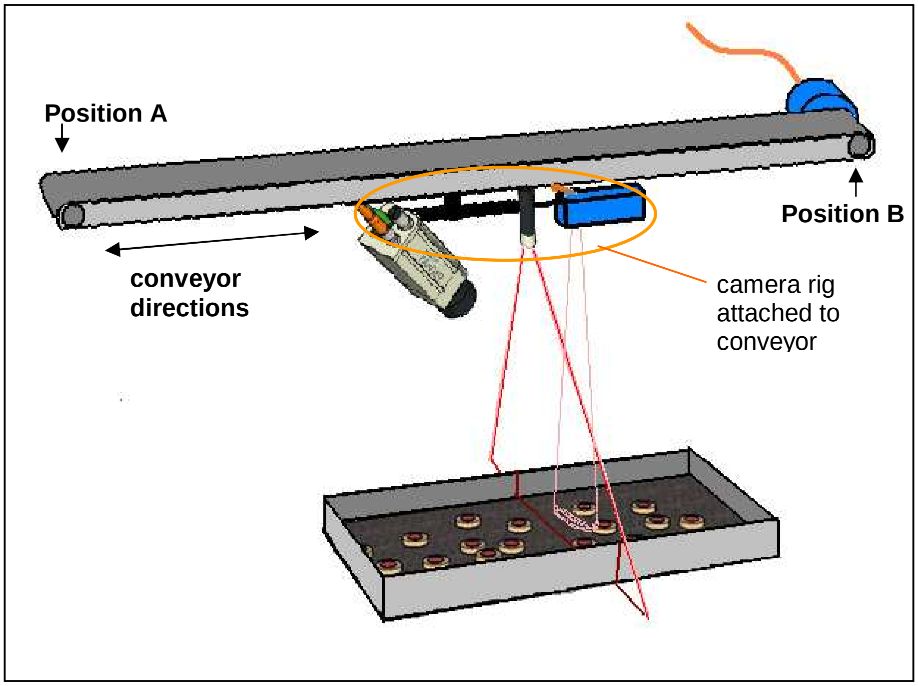


Figure 1.The object to be scanned is still and the camera and laser is moving. An encoder is connected to a linear motor and an enable signal is generated from a photo switch that follows the camera movement.

GENERAL

What ways are there to scan an object as seen in figure 1 above when the camera is moving and object is still?

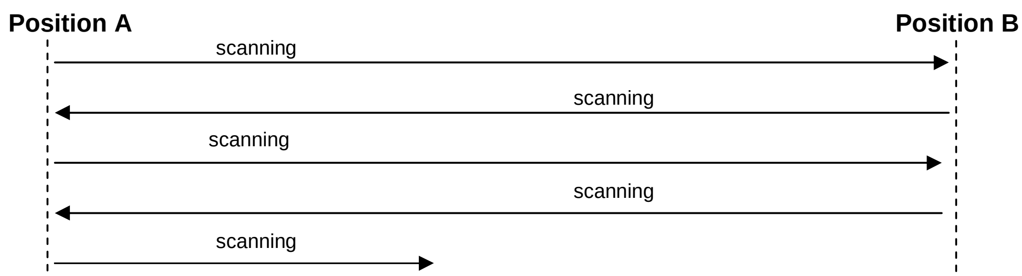
This document describes the different possibilities from an application point of view and discusses the pros and cons for each of them.

APPLICATION DESCRIPTION

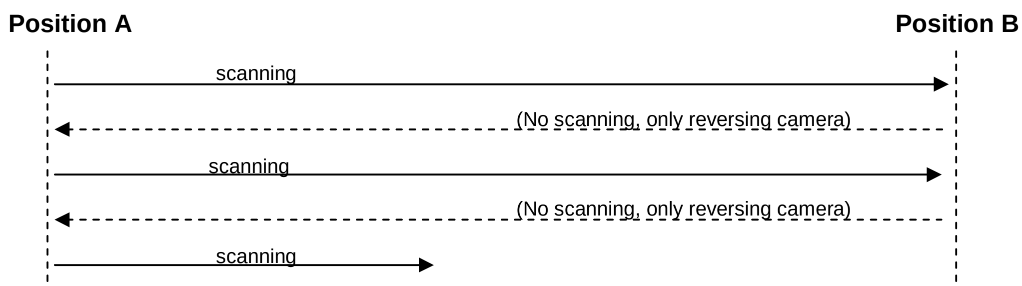
Continuous or repetitive scanning?

From an application point of view, given the scan setup seen in figure 1 above, there are really only 2 different ways of scanning the object(s). ***Continuous scanning*** and ***repetitive scanning***.

- In ***Continuous scanning*** the camera will acquire and send scans of the box and its contents to the PC in both forward and backward movement of the camera rig.



- In ***Repetitive scanning*** the box with its contents is scanned in one direction only, and then the camera is returned to its start position to start another scan sequence. During the return sequence no scans should be acquired.



Minimizing un-necessary 3D data in resulting PC buffer

For both these two scan modes, continuous and repetitive scan mode, normally only the data from the object (i.e. the box and its contents) is of interest. This means that the extra data before and after the box in the scan directions is probably not needed nor wanted from an image analysis point of view.

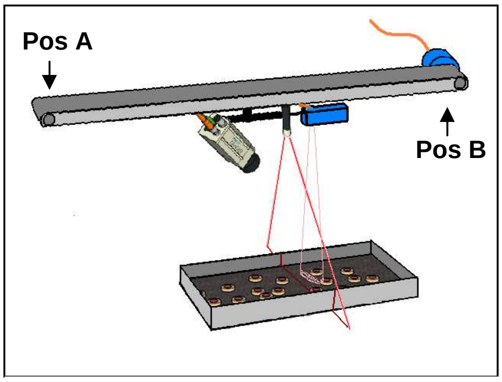


Fig 4. Record scene.

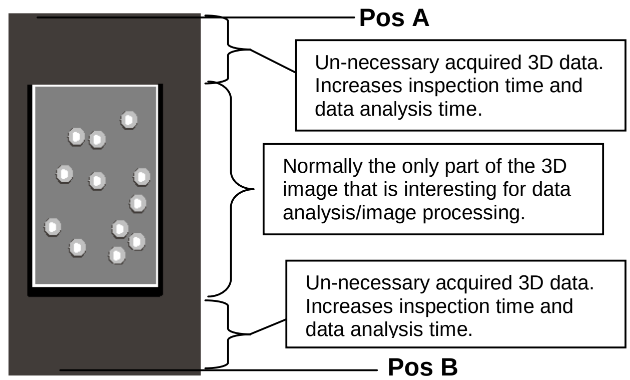


Fig 5: Fictive 3D result example of scanning the record scene in figure 4 from PosA->PosB. Colour coded height map: Bright is higher, darker is lower. Black is missing data.

- So we will try to minimize the extra data before and after the box in the resulting 3D image. This can be done by using an ***enable signal*** (normally a photo switch) to start camera scan acquisition, and also tell the camera how many scans to acquire for every enable signal. We will describe this process more in detail further down in this document.

Constant or non-constant conveyor speed?

Another aspect of the application is whether we can be guaranteed a constant conveyor speed in all positions over the object to be scanned, or if this might change during transportation of the camera rig

- Possible intermediate stops over the box?
- Acceleration and deceleration times affecting the constant speed over the box?

- If we have a constant conveyor speed we don't need an encoder to ensure that we get scans from the camera at constant scan speed with constant physical distance in between scans. The camera handles this anyway by using an internal counter instead of an external encoder. And the physical distance between the scans of the object depends on the constant conveyor speed.

- If we have a non-constant speed of the conveyor we need an encoder to get equidistant scans (scans with equal distance in between each other). And the physical distance between the scans of the object depends on the number of trigger signals per rotation of the encoder shaft.

Summary of application description

So, given the two scan modes of continuous or repetitive scanning we can combine the usage of encoder and photo switch(es) to minimize the resulting 3D data to more or less only hold the box and its contents and also we could be independent of any non-constant conveyor speeds.

How do we do this in the best way?

- Let's read the more detailed technical description of the enable and encoder functionality in the following chapter to understand how to optimize everything.

Technical description of the scanning system

Below we try to explain how to optimize some specific camera and PC buffer parameters in order to minimize the resulting 3D image and only record the wanted part of the record scene.

We also explain more in detail the available trig modes in the camera and the usage of the enable signal and their affect on the resulting data in the PC buffers.

Finally we enlighten some limitations of this scan setup and possible problems a user can encounter.

Photo switch usage (UseEnable)

In general using an enable signal for starting the scan sequence is a very good idea since you will have good control of when the acquisition starts due to the position of the object. This ensures you that you will only get data from the camera when there is an object present.

Photo switch usage for continuous scanning

For continuous scanning you can normally NOT use a *single* photo switch for starting the scan sequence. This is due to that the photo switch must detect the object before the laser line hits it, else we will loose the first 3D scans of the object (or in worst case the complete object).
If you want to use an enable signal in continuous scanning you need a logical system of two photo switches put together so that you get an enable signal before the laser line hits the object in both scan directions (PosA->PosB and PosB->PosA).

Photo switch usage for repetitive scanning

For repetitive scanning you "only" need to make sure the enable signal only goes high when you move the camera rig in one specific direction, in this example from posA to posB in figure 1 above. If the enable signal would go high on the reverse sequence of the camera rig, you would end up with much useless data from the parts outside the box since the photo switch detects the box only after the complete box has passed the laser line in the reverse direction.

You can avoid this problem by stopping the camera (stopping the scan acquisition) when reversing the camera, and restart the camera once back in start position again. However, stopping and restarting the camera takes extra time (1-2s), and you might want to avoid this since time is normally very valuable in any inspection system.

The number of scans sent from the camera at every enable signal

OK, let’s say we have a working enable signal that goes high when the camera rig approaches the object to be scanned (the box in this case). Then we can specify the number of scans the camera should acquire and send to the PC so that we don’t get too much data but only more or less the box itself in the resulting 3D image in the PC buffers. The number of scans is specified by the “scan height” parameter in the camera parameters, and this camera parameter is only valid when using an enable signal (i.e. when camera parameter UseEnable=1).

Assuming in this example below the box is 1000 scans long and we need some extra margin in the PC buffer, then we should set the Scan Height parameter = 1050 (or 1100 or something close to that).

The resulting 3D data we ideally would like to have in our PC buffer is probably looking like this:

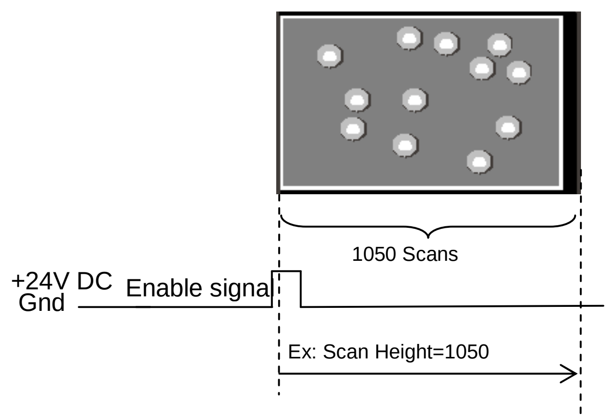


Fig. 6. Wanted result in PC buffer

We see that for every positive edge of the enable signal the camera will send “Scan Height” scans to the PC.
This means that if we set the scan height parameter to low we will never get the complete box scanned. And if we set it to too long, then we will get to much unwanted data after the box in the resulting 3D image in the PC buffer.

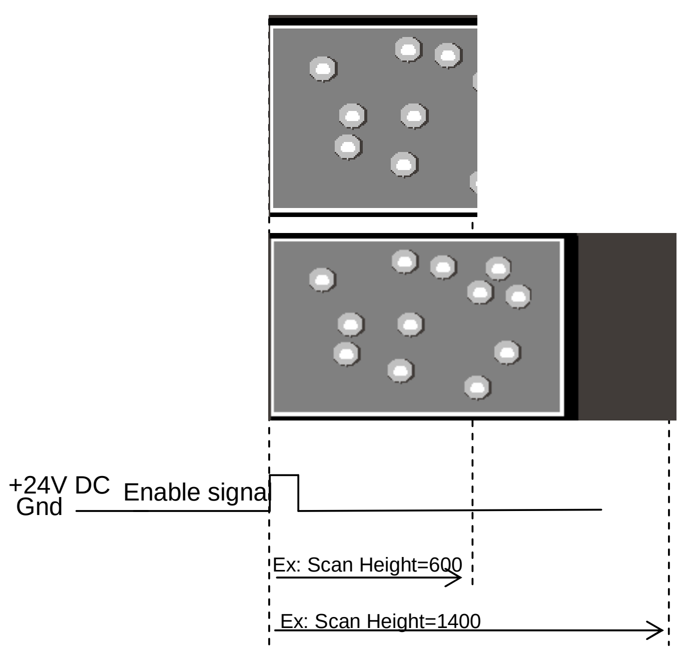


Fig. 7. Incorrect result in PC buffer due to non-optimized Scan Height parameter

How do we optimize the Scan Height parameter?

Well, first we need to understand the behaviour of the enable signal and the possible buffer sizes in the PC in order to understand ***all*** the pros and cons of setting the Scan Height to different values.

How does the camera interpret the Enable signal

The enable signal is implemented so that it reacts on the positive edge to start a scan sequence of “Scan Height” scans from camera to PC, and when this sequence is ready the current status of the enable signal is polled. If the status says “high” then another set of “Scan Height” scans are acquired and sent to the PC before polling the enable status again. And as soon as the polling detects the enable signal to be low, then only another positive edge of the enable signal can trigger the start of another scan sequence. This means that in between the pollings of the enable signal we can have any jitter on the enable signal without affecting the completion of the scan sequence.

Setting the buffer size in the PC

The camera itself has no knowledge what so ever about the size of the receiving PC buffer that holds the resulting scan data.

The total allocated PC buffer size is set by the user in his/her own PC application using the proper icon API command. The total PC buffer size is then automatically split into sub- (child-) buffers of wanted size depending on the data format from the camera and the buffer specific parameter “Lines per frame”

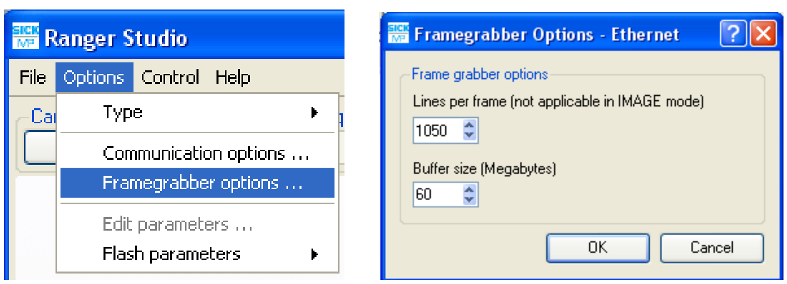


Fig. 8. PC buffer parameters seen from RangerStudio

Example: If the camera is set to deliver 3D data only with BYTE representation of each 3D value, then every scan will be 1536 Bytes long. Many consecutive scans with these 3D data are then used to build up the PC child buffer.
If the buffer settings are as in figure 8 above, then we have in total 60MB to use for data storage and every child buffer will consist of 1536x1050Bytes = 1575kB.
This will result in that we get 39 child buffers to put data in.

In the PC application the scan data will automatically end up in the next available child buffer and whenever this buffer is complete, the user can access this data by the data pointer provided by the icon api.

Note! This means that the data in a non-completed child buffer will not be able to access!

Optimizing the Scan Height value

Now when we know how the enable signal works we can discuss how we can optimize the Scan Height value.

We have three questions that need to be answered first:
Q1). Are the boxes to be scanned of identical lengths or do they vary in lengths?

Q2). Can we allow for always using a maximum image size to ensure the largest possible box to fit in the resulting 3D image in the PC buffer?

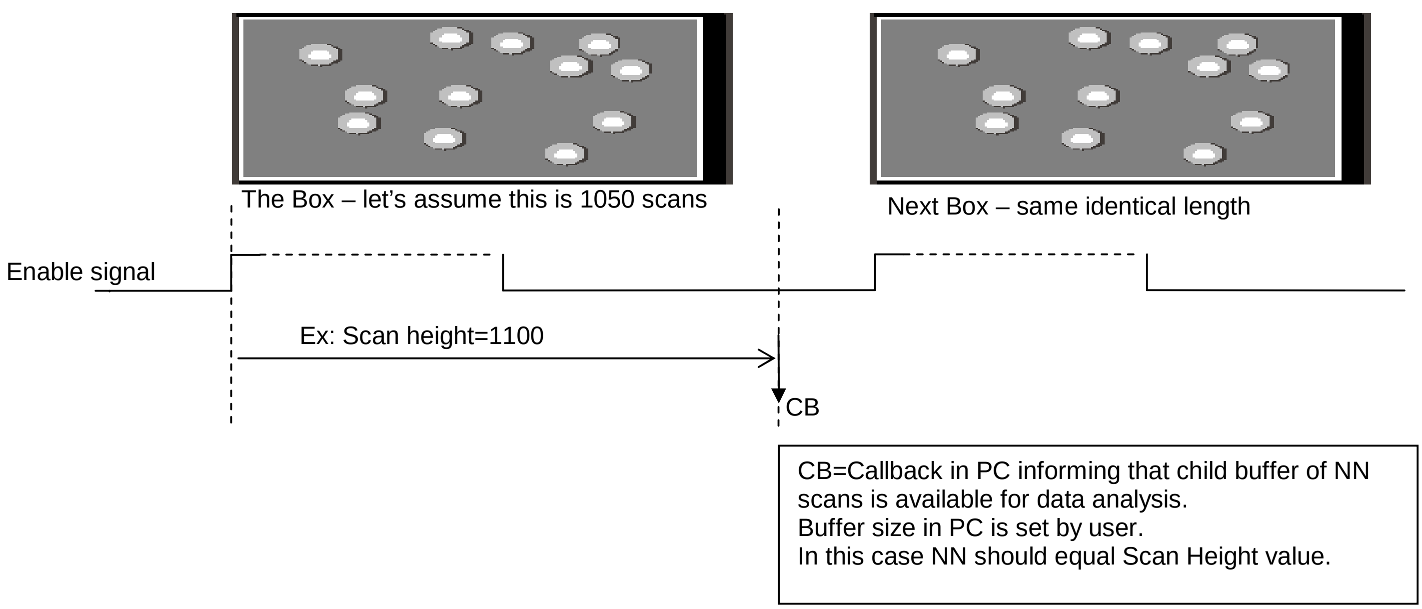
Q3). Is image analysis time a critical factor, meaning that we want to be able to start the image analysis of the first part of the box before we get the 3D data from the last parts of the complete box?

There are actually only two answers (A1 and A2) to the questions above: …

A1) If the boxes are of identical length and we can wait for the complete box to be scanned before we start image analysis …
…then we should use a large enough Scan Height parameter to always get the complete box scanned for a single positive edge from the enable signal.

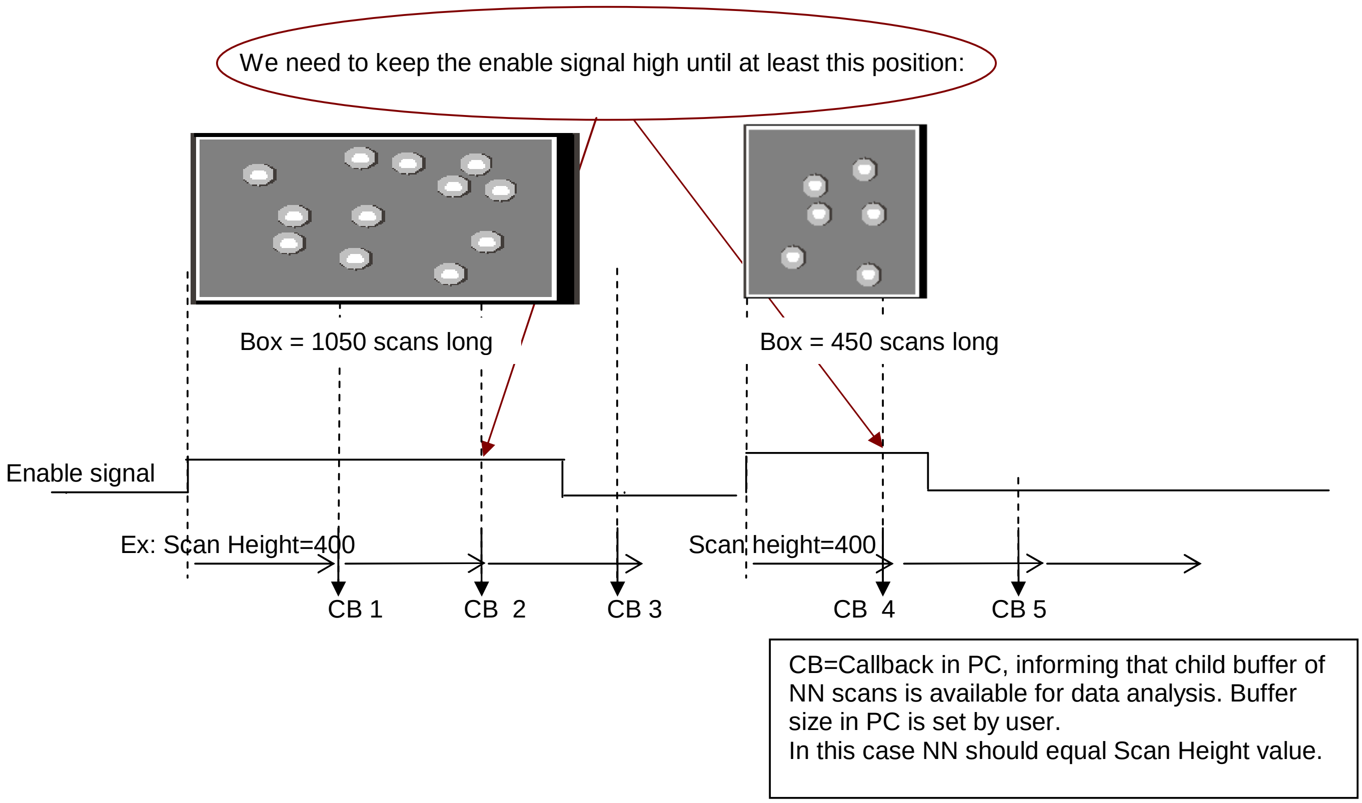
Make sure enable signal has gone low before scan sequence is completed, or else you will start another scan sequence …

There receiver buffer in the PC should be of equal length as the Scan Height parameter value!



A2) If the boxes are of different length or we don't have time for awaiting the 3D data from the complete box before we start image analysis…
…then we should divide the box up into smaller scan sequence chunks and also use several smaller PC buffers.

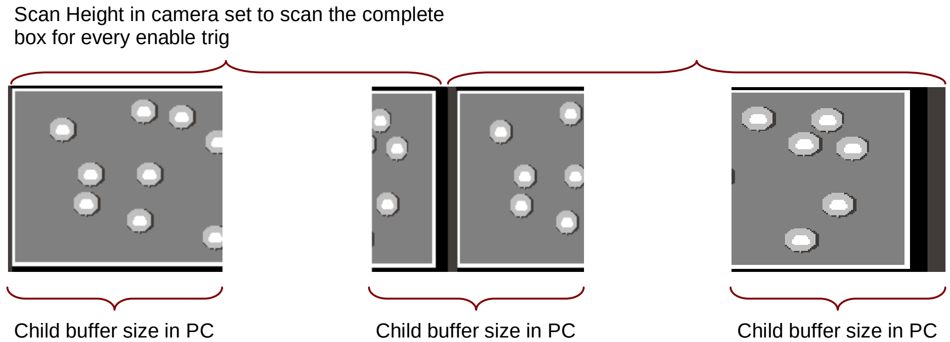
Important to know is that for this case we need to keep the enable signal high the whole time until the last scan sequence has been started, else we will loose scans of parts of the box.



We recommend the PC child buffer size to equal the Scan Height value, else there is no use of using smaller child buffers then the maximum possible length of the largest object.

At the very least we need the Scan Height parameter in the camera to be an even factor (x1, x2, x3, …) of the PC child buffer. Else we will loose synchronization and we will more or less always end up with non-completed buffers that will not fill up until the next positive edge of an enable signal is detected (which will probably be the start of a new object).

NOTE! This means that if we don't synchronize Scan Height parameter value with "Lines per Frame" parameter value, we will get the last parts of the previous object showing up in the same child buffer as the first part of the next new object.



Or even worse, if we are using repetitive scan mode and reversing the camera without stopping the scan acquisition - then we risk getting data from the same object showing up twice in the same child buffer!
This is because the camera will still be scanning since the Scan Height value is not reached and neither is the child buffer filled up so the user can get access to it. So changing scanning direction (to reverse the camera) will only result in that the end of the same object shows up again, but mirrored, in the same child buffer.

OK now we have talked about the Photo switch usage, now lets explain the different possible (encoder) trig modes in the camera

(Encoder) trig modes in the camera

We now know that for solving the application as described above in chapter 2 above, we would ideally need either continuous or repetitive scan mode.

Continuous or repetitive scanning can use either free running or encoder controlled scan acquisition.

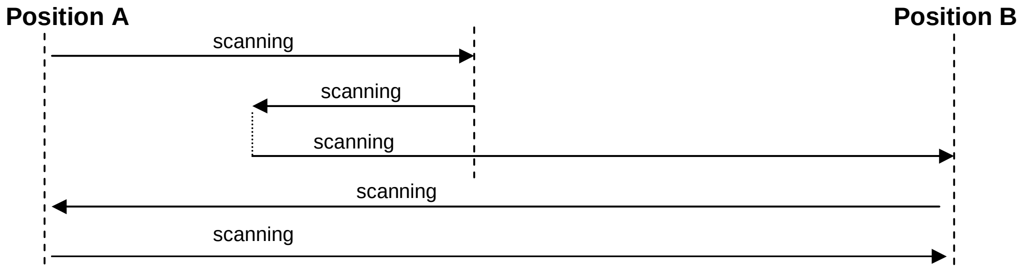
1) **(Free running)**
   The camera continuously acquires scans using internal cycle time control.
   This means that the camera will acquire and send scans to the PC even if the conveyor (camera rig) is standing still.
   *We **will not** use this trig mode in this application since we suspect non-constant conveyor movement*

2) **Encoder controlled**
   The camera acquires one scan per every n:th encoder
   Scans will only be acquired if the conveyor is moving, i.e. if the camera is receiving encoder trig signals.
   *We **will** use this trig mode to solve this application since we suspect non-constant conveyor movement*

From the discussion above in chapter 2.3 and 2.4 we will choose an encoder controlled acquisition and not a free running data acquisition in the camera.

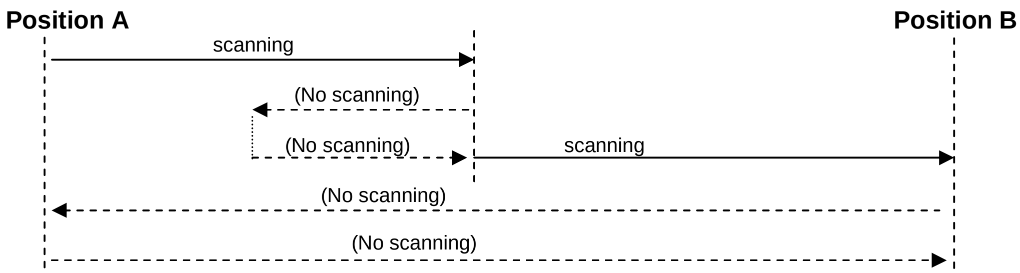
There are currently only two encoder controlled modes available in the standard camera SW.

- Trigger a scan on every n:th encoder trig regardless of conveyor direction, (known as “continuous scan mode”)



or

- Only trigger in one direction for every n:th scan, never scanning same position twice (known as “Log scan mode”)



From these modes we see that none of them fulfils our possible need from the application description (chapter 2), namely the repetitive scan mode. The repetitive scan mode is not implemented in the camera today.

We are thus forced to choose the continuous extern trig mode, i.e. trig in every direction of the conveyor.

Choosing the “Log scan mode”, we would only get data the very first scan sequence after staring the camera, then we would go back (reverse the camera rig) and try to scan same position again which the camera will not allow. Unless we stop the camera during reverse sequence and then restart the camera again for every new scan in the correct direction.

Even though repetitive scan mode is not supported specifically in the camera, we can achieve the needed functionality if we combine the “always trig on every n:th encoder trig regardless of direction” with a smart usage of Enable signals to only get enable in one direction of the conveyor (incl. camera rig).

Limitations of scan setup and potential problems

In this chapter we discuss some potential problems and possible suggestions/workarounds on how to avoid them.

Let us first assume we ALWAYS use the Enable signal to trigger the start of a new scan sequence, else we will have no control of when the data acquisition is started in the camera.

Assuming this we still can face some potential problems:

Changing direction of the camera rig during an ongoing scan sequence

Changing direction in the middle of an ongoing scan sequence in the camera will most probably result in a PC child buffer with incorrect data.

There are two alternatives:

1. Not changing the object but only the scan direction:
   This is likely to be the case when the camera is set to reverse too soon after having scanned in one direction only.

2. Changing the object as well as the scan direction:
   This is likely to be the case when we use the continuous scan mode and we are changing objects after one conveyor movement from position A->B or (B->A).
   and the camera rig changes direction too soon without having completed the ongoing scan sequence in the camera.

Either you will not get a complete box since we changed direction before we had scanned the last part of the box, or we end up with the last part of the box shown twice in the last PC buffer (the second time the same data is shown in the child buffer it will be mirrored due to refverse scan direction).

The only way you can cope with this potential problem is to always ensure that you let the camera finish the ongoing scan sequence (triggered by the Enable signal).
This means that in free-running mode you will probably not get into the same amount of problems since camera will send data regardless of conveyor speed (even when conveyor is standing still). In encoder triggered mode you will risk more problems since you will not acquire scans in camera unless you actually move the conveyor.

Stopping and restarting data acquisition during an ongoing scan sequence

If you have feedback from the mechanical conveyor so that you know that the object is scanned, then maybe you would like to get all the scanned data so far, including the ones in the on-going PC child buffer, since you know you are finished scanning your object.

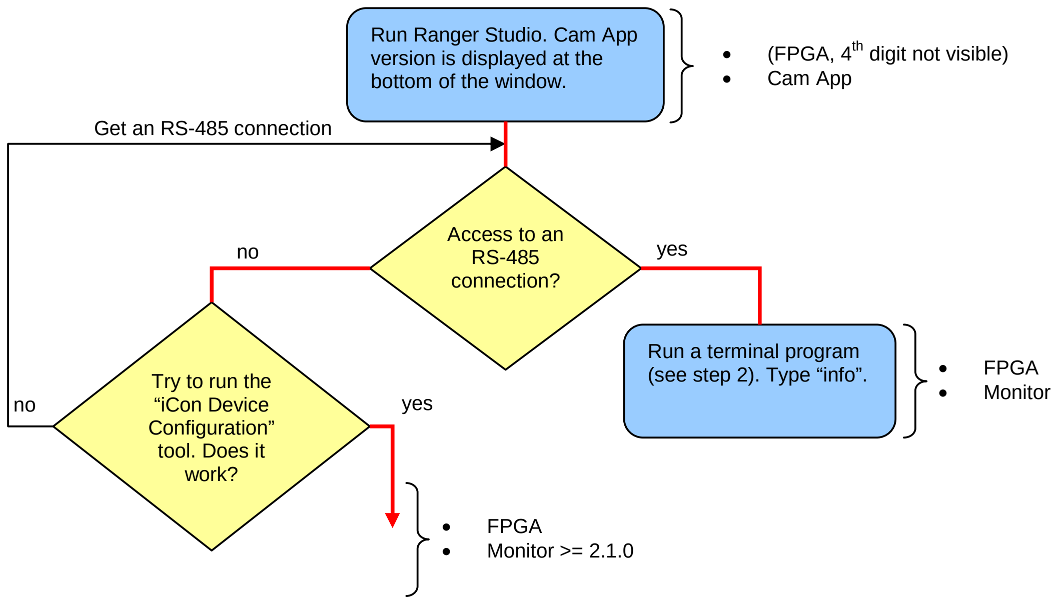
This data in the on-going PC child buffer is not so easy to get hold of (actually impossible today). You will be able to stop the ongoing data acquisition in the camera but then all the data in the on-going child buffer will be lost!

This means that you should always try to use Enable signal to the camera and make sure the child buffer size and the Scan Height parameter value equals. Then you know the camera will always fill up the last child buffer correctly. (that is, if the camera is not encoder controlled and the encoder doesn’t provide any trigs, i.e. conveyor is standing still).

Upgrade

U01 - How do I update the camera's firmware?

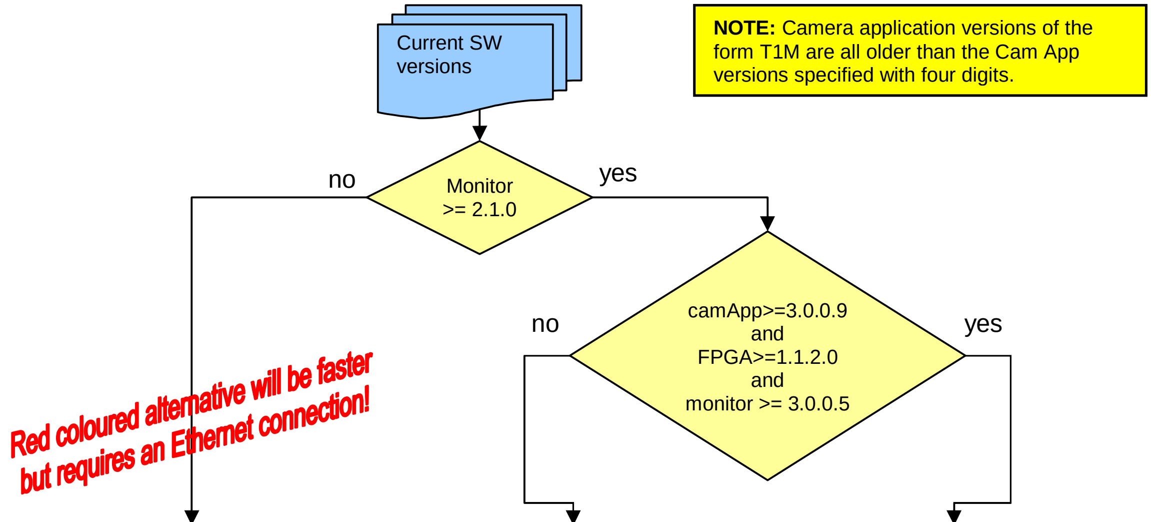
Step 1: How to get information about all different types of currently installed software



Either of the two alternatives, marked by red colour above, will guide you access to all necessary information about the currently installed software (FPGA, monitor and camera appl. version).

Step 2: Sort out possible SW/FW upgrade methods based upon the currently installed SW/FW

Follow the steps below to complete an upgrade.



	You may use: • **RS-485 (recommended)** **Alternative 1 in black** **Alternative 2 in red**	You may use: • **Ethernet (recommended)** • RS-485	You may use: • **iCon Dev Conf. (recommended)** • Ethernet • RS-485
INTRO	Set the Ranger in Monitor mode and start a terminal program (see section "How to set up RS-485 connection "). Set the Ranger in monitor mode, then start a terminal program. Check the IP of the device (if not known) using Ranger Studio or by configuring the fpga (key "c" in the terminal window) and then ethernet (key "i").	Set the Ranger in Monitor mode and start a terminal program (see section "How to set up an RS-485 connection "). **HINT:** If you do not know the IP-address of your camera please run the "iCon Device Configuration tool" and click "Update" to find it out.	Run the "iCon Device Configuration" tool. Click "Update" to find and select the device. **NOTE:** For this method the device shall be booted in normal application mode, NOT boot monitor using monitor ENABLE). It is strongly recommended to only upload a combination of Cam Appl and FPGA from the same 3D SDK release.
FPGA	* Type "load-fpga-file" in the terminal window. * File-menu -> "Send file". Type "efl" in the terminal window. In a PC command window, execute: *SendUDPPacket.exe <camera ip address> 5000 /FILE_VERSION1 FPGA\epm_top_struct<version>.hex*	* Open a PC command window, execute the command *SendEFL FPGA\epm_top_struct<version>.hex <camera ip address>*	* Select "Upload Camera Firmware". * Choose an FPGA (.hex) file to upload to the camera. DO NOT REBOOT YOU CAMERA UNTIL THE PROGRESS BAR IS COMPLETED. * Reboot the device.
MONITOR	* Type "load-monitor" in the terminal window. * File menu -> "Send file". Type "eml" in the terminal window. In a PC command window, execute: *SendUDPPacket.exe <camera ip address> 5000 /FILE_VERSION1 Monitor\EthernetMonitor_<version>.txt*	* Open a PC command window, execute the command *SendEML EthernetMonitor_<version>.txt <camera ip address>*	The Monitor can not be upgraded using iCon Device Configuration, instead it requires the Ethernet or RS-485 procedure described to the left.
CAM APP	* Type "pload32" on the RS485 connection. * File menu -> "Send file". Reboot when the command has completed. Type "epl" in a terminal window. In a PC command window, execute: *SendUDPPacket.exe <camera ip address> 5000 /FILE_VERSION1 CameraApplication_<version>.txt* * Reboot the device.	* Open a PC command window, execute the command *SendEPL CameraApplication_<version>.txt <camera ip address>* * Reboot the device.	* Select "Upload Camera Firmware" and choose a Camera Application (.txt). * Reboot the device.

DO NOT REBOOT YOUR CAMERA UNTIL THE PROGRESS BAR IS COMPLETED, about 20 sec.)

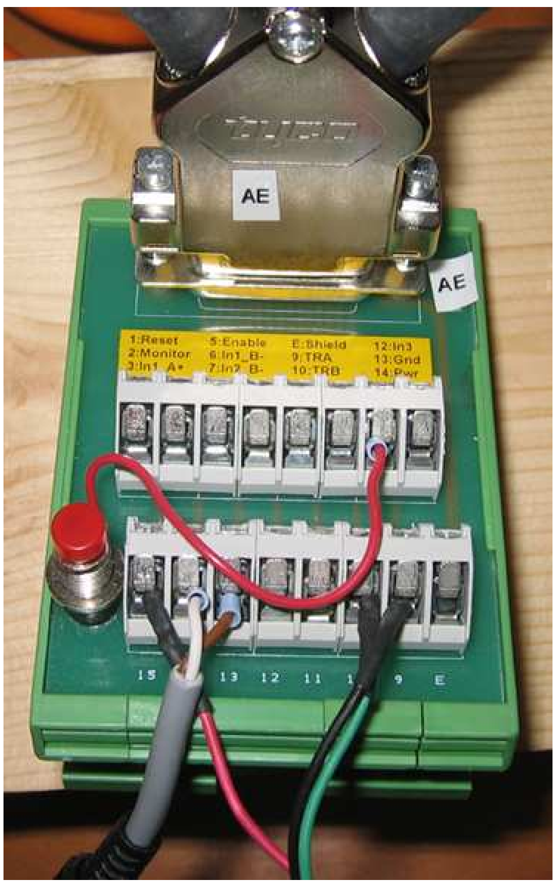
(Optional) How to set up an RS-485 connection

Connection (for more information see the Reference Manual)

- The red wire linked to the red button should be connected to pin 2 (monitor) on the green I/O extension box.
- The power supply cable should be connected to the pins 13 (Gnd) and 14 (Pwr).
- COM connection runs through the pins 9 (TRA), 10 (TRB) and 15 (Gnd).

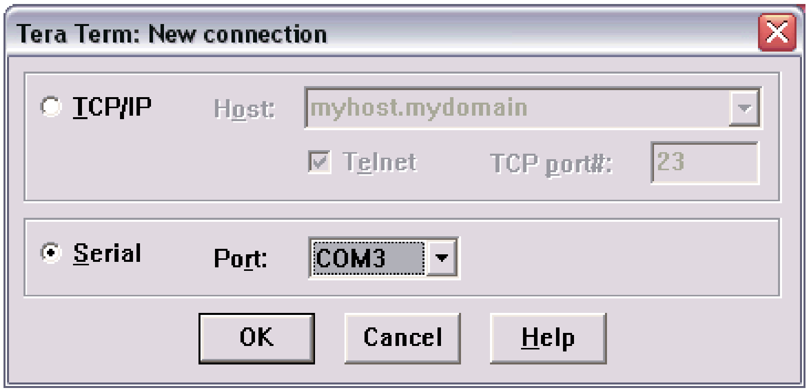
To enter monitor mode do the following:

1. Hold down the red button
2. Power up the camera
3. Release the red button (the red button can be released as soon as the power is connected)

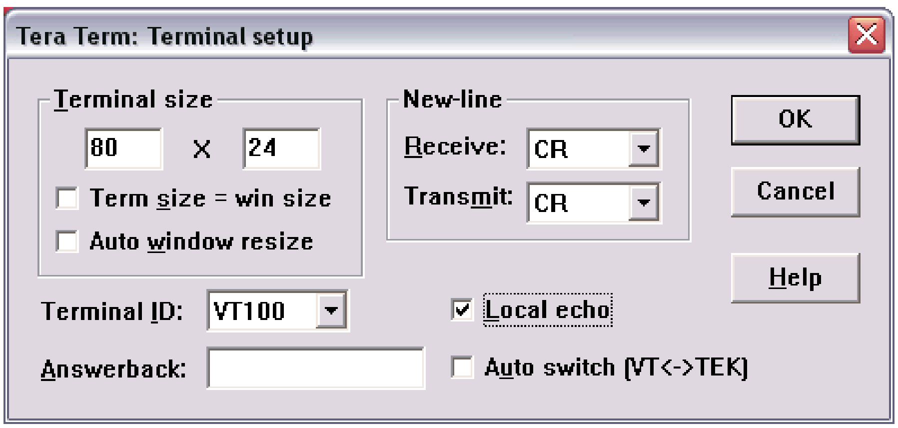


Step by step tutorial in the terminal program "Tera Term"

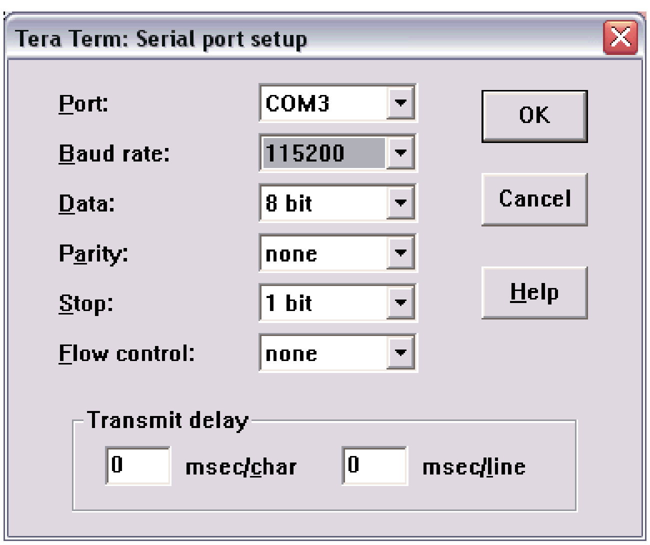
1. Start Tera Term
2. Select COM port (COM1, COM2, COM3, COM4) -> OK



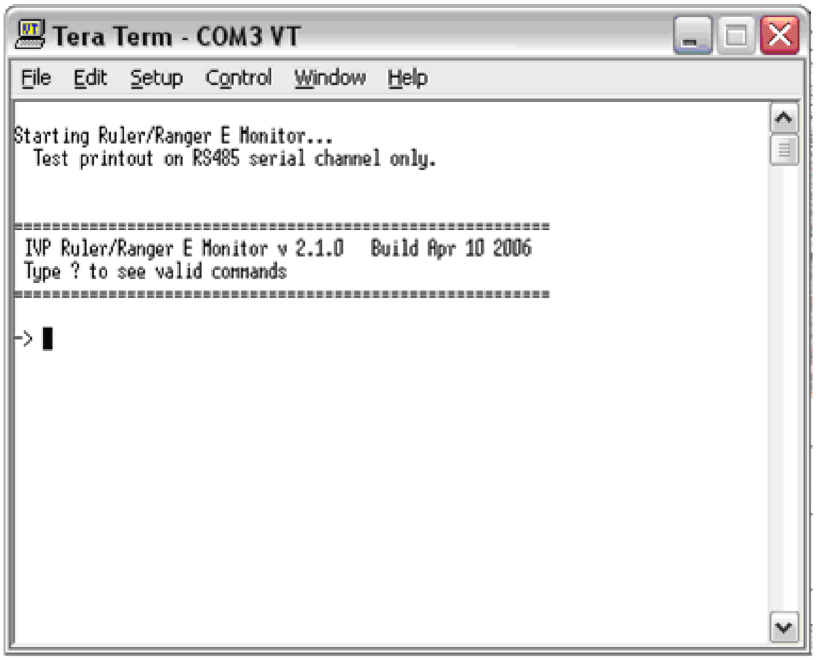
3. In Tera Term: Setup -> Terminal -> click "Local echo" check box -> OK



4. Setup -> Serial port -> select baud rate, 115200 -> OK



5. Disconnect power from camera -> hold in the red button on the terminal board and connect the power. If connection is made OK, then you should see some similar printouts as in the image below.



6. In the image above you can see that the camera used in this example has got monitor 2.1.0 installed. To show all possible commands from within monitor mode type "?" and hit the enter button.

U04 – What is the difference between the monitor, fpga and camera application? Which ones do I need to update?

The monitor, FPGA and camera application exist in the camera flash as separate files and it is therefore possible to have different combinations of these files in the camera.

Monitor

The monitor includes functions to configure the camera, update the monitor, fpga and camera application. These functions are only accessible when the camera is in monitor mode. It is possible to set the camera in monitor mode while booting the camera.

Camera application

When the camera boots without entering the monitor mode, the camera application will be entered. It is in the camera application that the functions to control the camera and the measurements exist. In camera application mode it is possible to start and stop the camera, acquire data etc.

FPGA

The camera application and the monitor uses the fpga for several tasks such as i/o information, measurement calculations etc.

Update

It is recommended to update both the fpga and camera application when there is a new release. The reason for this is that during the release testing phase, the fpga and camera application combination is tested and it is only this combination that is verified to work properly.