



In-Sight 3800L Installation Guide

5/1/2025
v1.1

Version History

Version History

The following updates were made:

v1.1 – 1 May 2025

New

Firmware 25.1.0 (and higher) now enables:

- The Integrated M12 light port on the back of the IS3800L
- Reduced Image mode
- Encoder-based delays

CONTENTS

CONTENTS

1	Introduction	1
2	Line-Scan Vision-System Basics	2
3	Applications for In-Sight Line-scan systems.....	4
4	In-Sight 3800L Specs	6
4.1	Raw Specs	6
4.2	Example Scenario	6
5	Lensing Options	7
6	Lighting Options	9
6.1	IO Module.....	9
7	Triggering Options.....	10
8	Clipping Modes	11
9	Encoder based Delay and Trigger Queueing.....	14
10	Encoder.....	15
11	Troubleshooting Line Scan.....	22
Appendix A	System Alignment Techniques	A-1

Introduction

1 Introduction

The In-Sight® 3800L vision system uses line-scan image-acquisition technology to build an image. This is unique from the more common area-scan technology used in cameras. Line scan offers a dynamic image size that is created during movement. The area scan is limited to capturing the same size image in a “freeze frame.”

Given the nature of movement required to build a line-scan image, there are unique considerations when integrating these systems as compared to area scan. This document is a practical guide for setting up a line-scan system as well as go over the specifics and setup of the In-Sight 3800L for use in your vision solution.

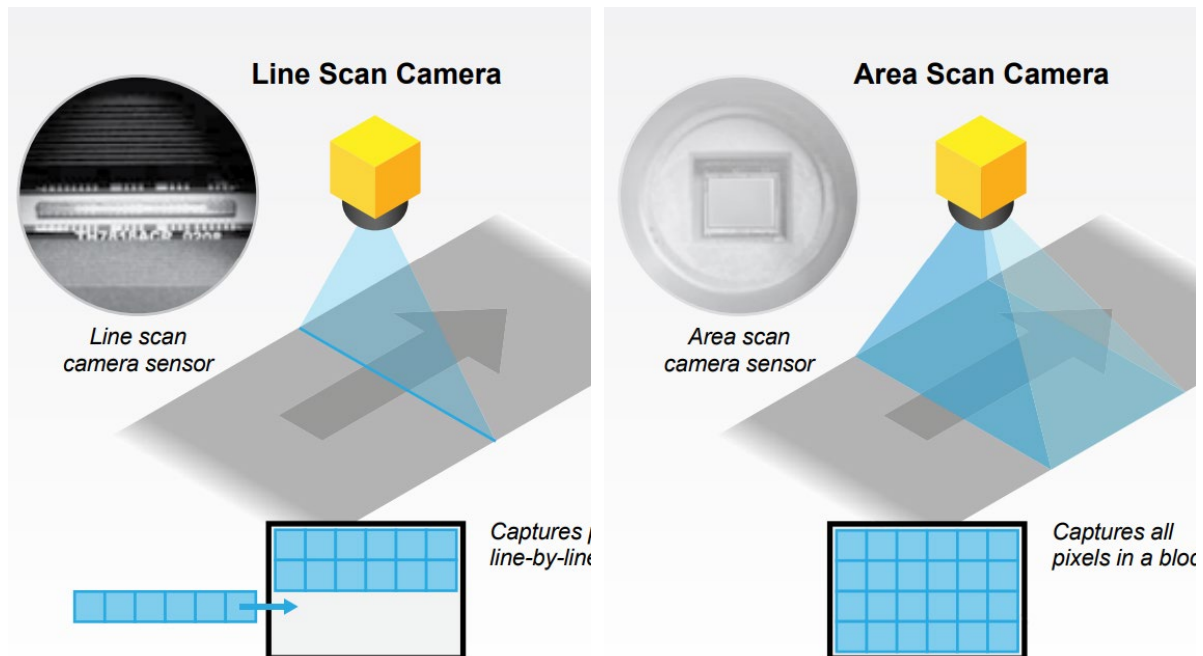
Line-Scan Vision-System Basics

2 Line-Scan Vision-System Basics

Line-scan vision systems build an image line by line and use the movement of the object being scanned as the trigger to acquire the next line. These are the key advantages compared to area scan:

1. The image size can vary depending on the object size.
2. The placement of these systems can be more advantageous when there is very little line of sight to the object being imaged such as a bottom side of a package moving between two conveyors.
3. Line-scan systems can typically splice in a previous portion of the last image onto the currently acquired image to ensure no gaps in data between full image acquisitions making them an ideal candidate for continuous web applications.
4. When an item needs to be unwrapped into a full image, the line scan can do this very well without distortion. This is very common when analyzing labels printed on cylindrical items such as can or bottle labels.
5. When an application needs to have a high resolution, the line scan can have a very high resolution horizontally (2k, 4k, 8k, 16k) with the Y-direction being a function of acquisition speed relative to the object speed.

Line-scan systems accomplish this by having very few rows of pixels (mostly one) in the sensor to read in each line. These lines are read from very quickly resulting in a far greater acquisition speed compared to an area scan.



Line-Scan Vision-System Basics

With these characteristics of line-scan systems, there are key aspects to account for:

1. Alignment of the system is critical for optimal image formation.
2. Lighting needs to be more focused and typically brighter than traditional lights as the exposure times will be significantly shorter than traditional area scan.
3. To accurately track the part, encoders are needed to handle variances in part speed as slight changes in speed can drastically affect the image formation if no encoder is used.

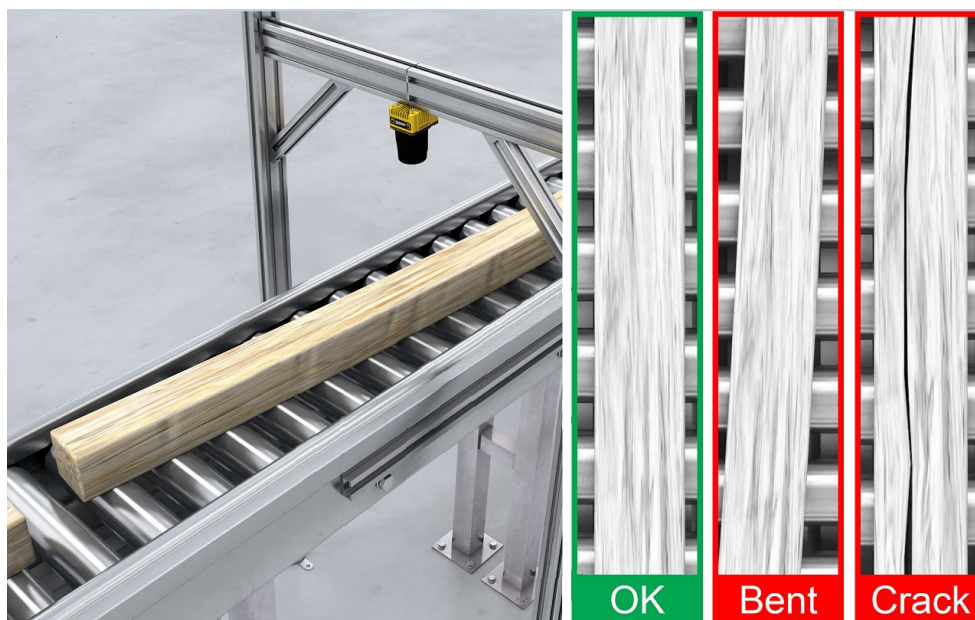
Applications for In-Sight Line-scan systems

3 Applications for In-Sight Line-scan systems

Since line-scan acquisition builds an image line by line, only a sliver of the object you want to inspect needs to be visible at any given time to the line scan camera. As the product moves past the camera, new lines are acquired until the full image (total number of lines) is built to reveal the full product view. This presents several advantages over area scan camera, as illustrated in the following application descriptions.

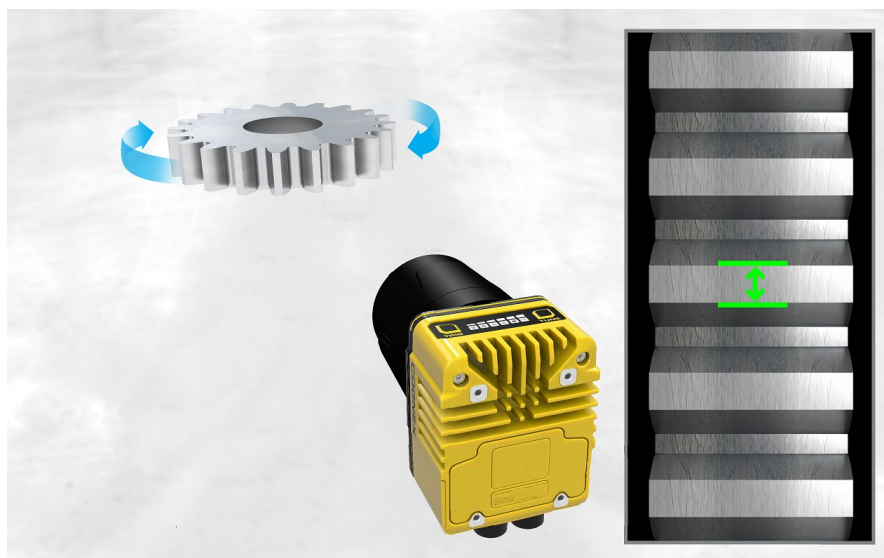
Large Parts

Parts several times longer than they are wide can be good candidates for line-scan systems as the image length is variable. Parts such as poles, lumber, flooring, make for good applications.



Cylindrical Parts

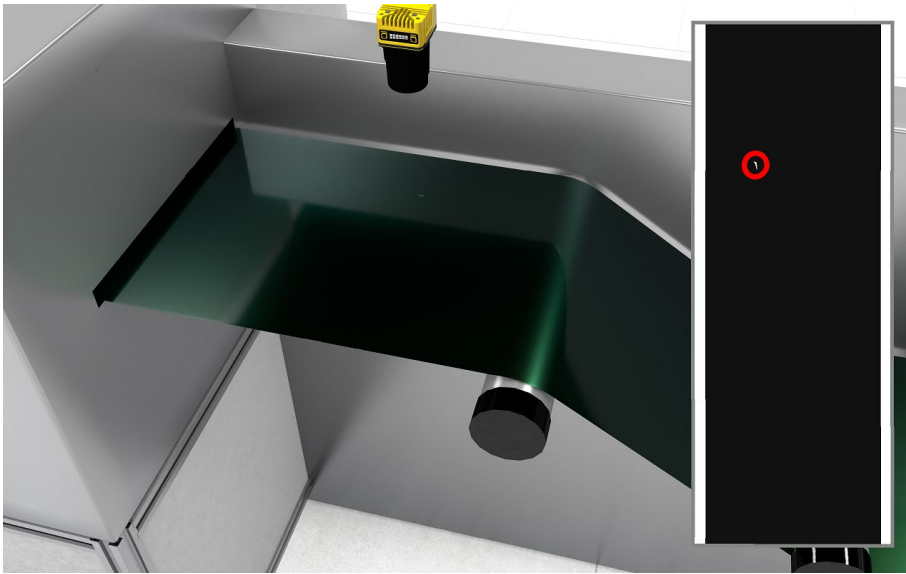
Because line scans take only a sliver of an image at a time, cylindrical parts can appear flat and easier to inspect without appearing distorted due to the shape of the object. Inspections for can labels, bottles, gears, tires, and other round objects can make for good candidates.



Applications for In-Sight Line-scan systems

Continuous Material

Continuous material applications like electrode sheets, carpet, linen, and other continuous web-based applications are good candidates for line-scan systems. With overlapping ability in the In-Sight 3800L, you can ensure that defects are always able to fully present when they might be in between image slices.



In-Sight 3800L Specs

4 In-Sight 3800L Specs

4.1 Raw Specs

Imager

Specification	1K Mode	2K Mode	4K Mode
Mount Style	C-Mount only	C-Mount only	C-Mount only
Sensor Type	1" CMOS	1" CMOS	1" CMOS
Bit Depth	8-bit monochrome	8-bit monochrome	8-bit monochrome
Line Rate (Max)	87 kHz (11.5 μ s per line)	169 kHz (5.9 μ s per line)	84 kHz (11.9 μ s per line)
Sensor Pixel Size	14 μ m x 14 μ m square pixels	7 μ m x 7 μ m square pixels	3.45 μ m x 3.45 μ m square pixels
Max Resolution	1024 x 16,376 (16 MPx)	2048 x 16,376 (32 MPx)	4096 x 16,376 (64 MPx)
Max Shutter Frequency	0.1 μ s to 1,000,000 μ s	0.1 μ s to 1,000,000 μ s	0.1 μ s to 1,000,000 μ s

Performance

- Memory: 4GB
- Image Processing Memory: 768 MB SDRAM
- Storage Memory: 7.2GB non-volatile flash memory.

IO

- Discrete Inputs: 1 Opto-isolated, acquisition trigger input, 2 dedicated encoder line inputs for quadrature support. 1 additional general-purpose input.
- Discrete Outputs: 2-Highspeed outputs.
- Encoder Specification: 12-30 VDC (500kHz max input rate with up to 15m long cables).
- Network: 2-Ethernet X-Coded ports. (Eth-0 supports Industrial Comms: TCP/IP, PROFINET, EtherNet/IP, SLMP, ModbusTCP, (S)FTP, RS-232C, IEEE 1588 (CIPSync))
- Power Consumption: 24 VDC $\pm$ 10%, 2.0 Amp maximum.

4.2 Example Scenario

In this document, we will go through the process of designing our line-scan system to meet the requirements of an example setup. In this setup, we will be looking at maintaining a minimum 200mm wide field of view with a 400mm working distance. The speed of the system is 650ft/min. Using this as our design criteria, the following sections will go through choosing the proper components and system settings for our line-scan system using the In-Sight 3800L.

Lensing Options

5 Lensing Options

The In-Sight 3800L has a 1 in. C-Mount only sensor size and lensing options chosen should be of the same type to avoid distortions.

Since the width of the line scan sensor is the gating factor in the field of view, determine the appropriate lens size using the below equation. The actual CMOS width is: 14.131mm.

$$Focal\ Length_{width} = \frac{Working\ Distance * Sensor_{width}}{Field\ of\ View_{width} + Sensor_{width}}$$

For example, what is the focal length of the lens needed to maintain the 400mm working distance with a 200mm wide Field of view?

$$Focal\ Length_{width} = \frac{400mm * 14.131mm}{200mm + 14.131mm} \approx 26.4mm$$

Since this isn't a very common focal length, a more common 25mm would be a more available option.

$$Field\ Of\ View_{width} = \frac{Working\ Distance * Sensor_{width}}{FocalLength} - Sensor_{width} \approx 212.0mm$$

Once the lens is selected, the image resolution and pixel size can be determined. The In-Sight 3800L sensor width is 4096 pixels; use the following equation to determine the resolution of the acquired image.

$$Resolution_{mm/px} = \frac{Field\ Of\ View_{width}mm}{Sensor\ Pixel\ Count\ px}$$

Using the calculated FOV from the last example of 212mm and the pixel count from the In-Sight 3800L of 4096 pixels, we get the following.

$$Resolution_{mm/px} = \frac{212mm}{4096px} \approx 0.052 \frac{mm}{px}$$

Once this is known, we want to "square" our resulting pixel sizes in our acquired image. Ideally, pixel size should be in a 1:1 ratio (width = length) to maintain square pixels in the final image. Since line scan imagers build the image one pixel line at a time, each line that is captured is the ideal pixel length. The field of view length can then be determined from the resolution.

$$Field\ Of\ View_{Length} = PixelSize * Number\ Of\ Acquired\ Lines$$

For this example, assuming square pixels, your maximum field of view length would be

$$Field\ Of\ View_{Length(max)} = 0.052mm * 16,376lines \approx 847.5mm$$



Note: The process for squaring the pixels will be further explained in the [Encoder](#) section of this document. For now, we will assume that acquisition settings have been set up appropriately to enable the 1:1 ratio.

Lensing Options

Below are some recommended Moritex lenses that are paired to work with the In-Sight 3800L camera.

Lens	Part Number
6mm SR Series	ML-U0618SR-18C
12mm SR Series	ML-U1217SR-18C
16mm SR Series	ML-U1615SR-18C
25mm SR Series	ML-U2515SR-18C
35mm SR Series	ML-U3518SR-18C
50mm SR Series	ML-U5022SR-18C

Lighting Options

6 Lighting Options

Because the In-Sight 3800L line-scan imager is a single row of pixels, only a single line of light is needed. This not only minimizes the space needed to illuminate the part but also means the entire part does not need to be uniformly illuminated – just the single line being acquired. In general, line-scan cameras need higher intensity light than area-scan systems due to the short acquisition speed.

A line-scan acquisition is typically in the low microsecond range, as opposed to the millisecond range of area-scan cameras. It is important not only to expose high-intensity light to the line-scan imager, but to also make sure the line light is even. As long as the light is even across the part in one line field of view, the entire image (4096 x 16376) will have uniform lighting.

Line-scan lights are typically high-intensity LED bar lights that converge the light to a small width, thereby increasing the intensity of the light in one line. Typically, these lights will have their own specific lensing to better focus the light as compared to other wider-angle bar lights.

Below are some recommended light models to consider:

Brand	Model Series	Notes
Moritex	CV-LN	50-150mm wide. White light only. Low-Medium intensity.
Moritex	MLNC	100-300mm wide. White/Red light options. Low-Medium intensity.
EffiLux	CCS EL	200-300mm wide. White/Red/Blue light options. Medium intensity. Bar light that converts to line scan light with kit (ICELB-CL-x00-KIT)
EffiLux	Effiline3	Up to 3,000mm wide. Various colors. Medium-High intensity.

For installations that do not require more than 18W (750mA max at 24 VDC) and that are running firmware 25.1.0 or higher, you can now use the integrated M12 connection on the back of the IS3800L to connect directly with your light for simpler applications.

If you find that your lighting is still not intense enough and you don't necessarily need the resolution, consider trying 1K mode. In this mode, your effective pixel area (14 μm x 14 μm) is quadruple that of 2K mode (7 μm x 7 μm) which allows for more light per pixel, helping you in darker illumination situations.

6.1 IO Module

At the time of this writing, the In-Sight 3800L does not support the legacy CIO-1400/Micro-based I/O modules. If you need to use more I/O than the I/O available through the breakout cable, please investigate using the Gardasoft CC320 TCP/IP model. This model can be integrated through spreadsheet snippets and can allow for an additional eight discrete inputs and outputs.

Triggering Options

7 Triggering Options

Triggering the system is very similar to any traditional In-Sight system. The big difference is that the encoder is also used to make sure the line-by-line acquisition takes place. The In-Sight 3800L has the following options when it comes to triggering.

Single

Single mode starts the image acquisition process at the start of each trigger signal it receives. This signal can either be at the input0 of the camera, trigger button (top of camera), network (UDP), factory protocol (Ethernet/IP/Profinet/Modbus/etc.), or even the WebHMI. With each trigger received, the camera will initiate a new image and will acquire new lines based on either hardware encoder pulses or line period of a software encoder until the number of acquired lines matches the desired image size.

Self

In self-mode, the camera will trigger based on the trigger interval set. When the interval is met, it will continually acquire lines based on the line period until the full image is built. It will not retrigger until the trigger interval is met. This mode does not support overlapping or clipping and thus will simply create an image based on the set image length.

Hardware Trigger

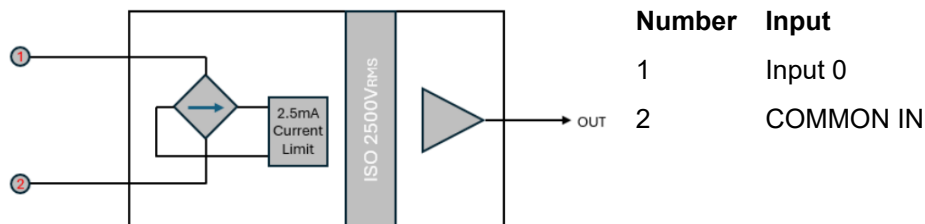
The In-Sight 3800L has one hardware input for the trigger (Input-0), which is optically isolated, that can be setup to either by current sinking (NPN) or current sourcing (PNP). Note that the voltage range is 12 VDC to 30 VDC with a 2.5mA limit. The maximum triggering frequency is 500 kHz when using a Cognex 15-meter cable or shorter.



Note: The COMMON IN does not refer to the typical electrical Common/Ground connection. It's meant to refer to the connection that is shared between the input(s) and/or outputs. This means that if it's connected to 0V/GND, the input will be expected to be HIGH/ON when voltage is applied. Conversely, if COMMON is connected to 10-30 VDC, then the input will be HIGH/ON when the input has a path to 0V/GND. See below for the connection options.

To trigger from an NPN type photoelectric sensor or PLC output, connect COMMON IN to +24 VDC and connect IN 0 to the output of the photoelectric sensor. When the output turns ON, it pulls TRIGGER down to 0 VDC, turning the opto-coupler ON.

To trigger from a PNP photoelectric sensor or PLC output, connect IN 0 to the output of the photoelectric sensor and connect COMMON IN to 0 VDC. When the output turns ON, it pulls TRIGGER up to +24 VDC, turning the opto-coupler ON.



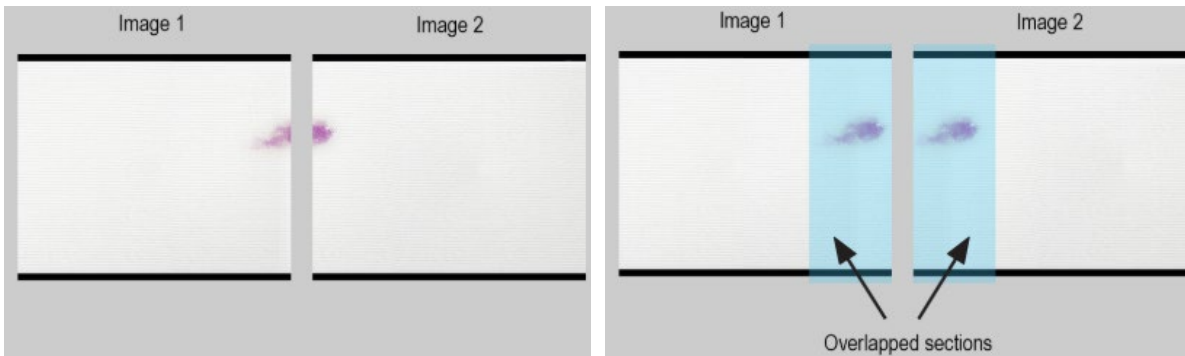
Clipping Modes

8 Clipping Modes

The In-Sight 3800L has ways of handling what happens if a new trigger is issued to the camera before the full image was finished building.

Overlap

In Overlap mode, if the trigger for the next image comes in before the current one is finished being acquired, the remaining portion from the previous image from the last acquisition will be appended to the current image. This is helpful when you have a defect that spans two images but needs to be fully in the inspected image to get a reliable result.



Without Overlap: Defects or features can be split between two images with gaps between them.

With Overlap: Defects or features are never split between two images with no gaps in coverage.



Note: This function requires that the trigger timing be such to obtain a desired overlap. For example, if you have a desired full image size of 1548px of current acquisition with the previous 500px of the last image appended, you would need to set your desired image length size to 2048px and trigger the camera every 1548 line periods.

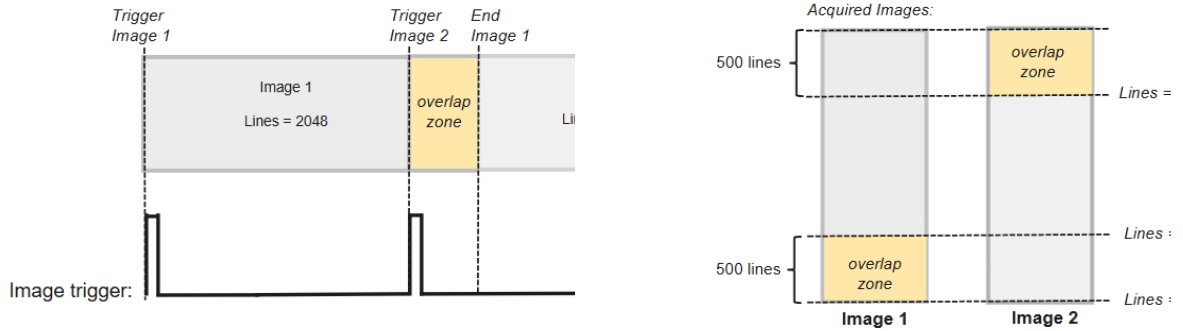
For example, if you wanted your image size total to be 2048 and your line period rate on the line scan is 20μs per line, then the camera would need to be triggered every 31ms to append the remaining 500px from the last image.

$$Trigger\ Rate_{ms} = (full\ image\ length_{px} - overlap_{px}) * line\ period_{ms}$$

$$Trigger\ Rate_{ms} = (2048\ px - 500\ px) * 20\mu s \left(\frac{1\ ms}{1000\ \mu s} \right) = 31\ ms$$

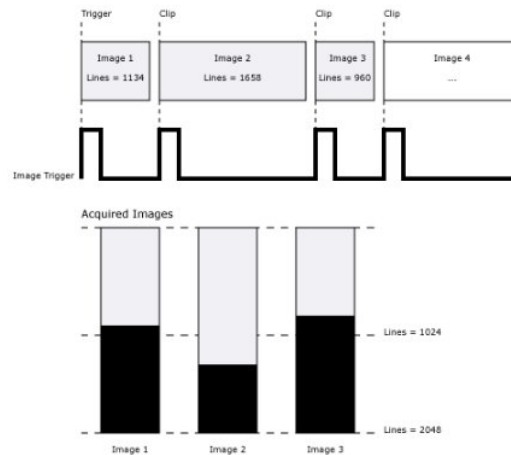
Clipping Modes

The diagram below shows how the overlap is taking place.



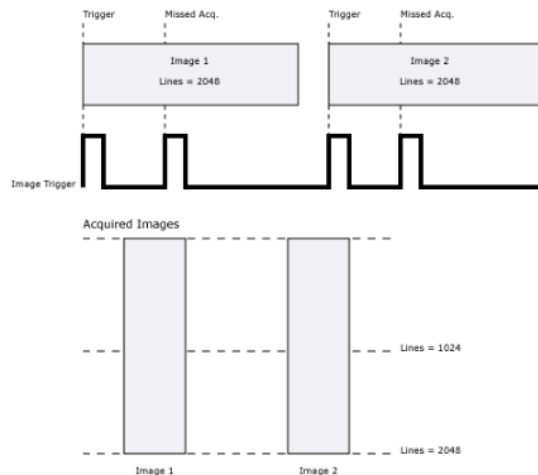
Fill Black

Like overlapping, this will append black pixels into any remaining image length that was not used and immediately start the next image upon receiving the trigger. This keeps the image size consistent. Unlike overlap, the second image does not contain any data from the previous image.



No Clipping

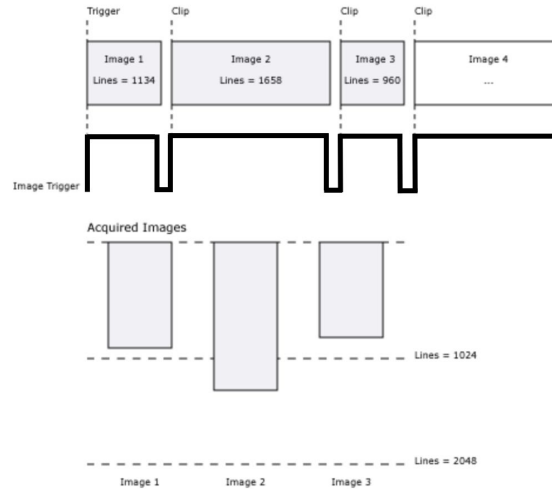
In No Clipping mode, any triggers received before the image isn't built will be ignored. This is shown with the diagram below where two out of the four triggers are ignored because the image was still being built when they arrived resulting in two full images opposed to four images.



Clipping Modes

Reduce Image Mode

Starting with In-Sight Vision Suite (ISVS) 25.1.0, the Reduced Image mode is available to use. This allows you to create a variable-sized image up to your max image length. This is best used in conjunction with a photo eye for variable-length-based parts.



Note: This function requires that the trigger be set to HIGH during the entire length of the image acquisition. Your acquisition duration will be set to “End of Trigger” automatically in ISVS and cannot be changed.

Encoder based Delay and Trigger Queueing

9 Encoder based Delay and Trigger Queueing

With the latest release of ISVS 25.1, you can enable encoder-based delays. This allows you to specify the number of encoder ticks between when the trigger goes high to when the acquisitions take place.



Note: Note that if using reduce image mode, the acquisition will remain for the same delay amount after the trigger goes low to keep the same image height as intended.

If the gap between the triggers is smaller than the encoder-based delay gap, the IS3800 can now queue the triggers so that the camera is able to maintain the correct image acquisition at the right time. The limit of triggers that can be queued is 18.

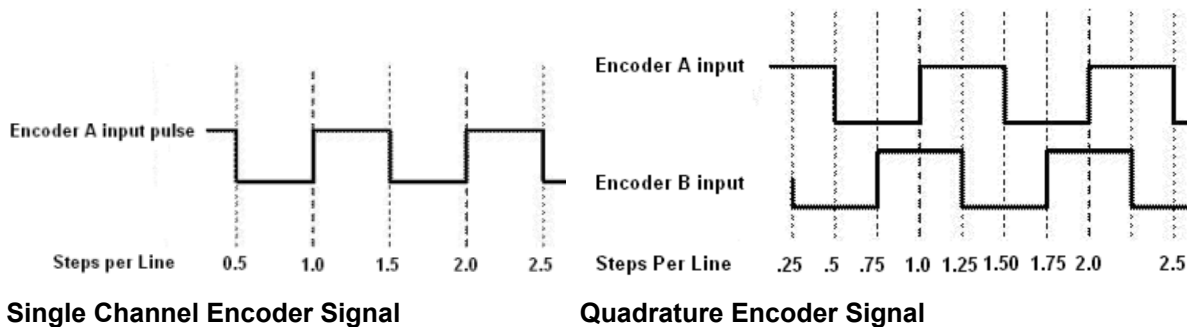
Encoder

10 Encoder

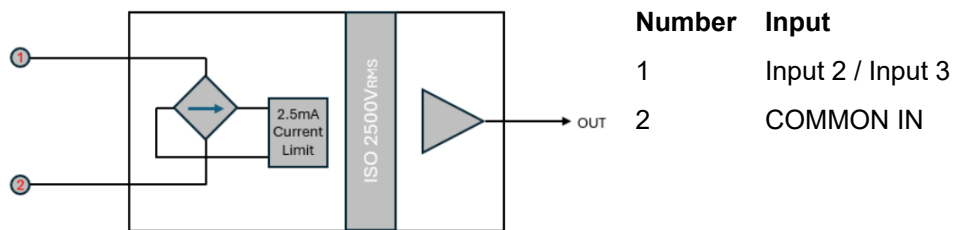
The In-Sight 3800L supports both hardware and software encoder modes. When the speed of the system can change, a hardware encoder is the best option as it can account for any variances in speed and keep the image formation consistent. If the speed of the system is very consistent and not expected to change during the inspection, a software encoder can be used to trigger the line scan at time-based intervals. Even with a system that appears to be constant, minor changes in speed can really alter the image formation in a line-scan system.

Hardware Encoder

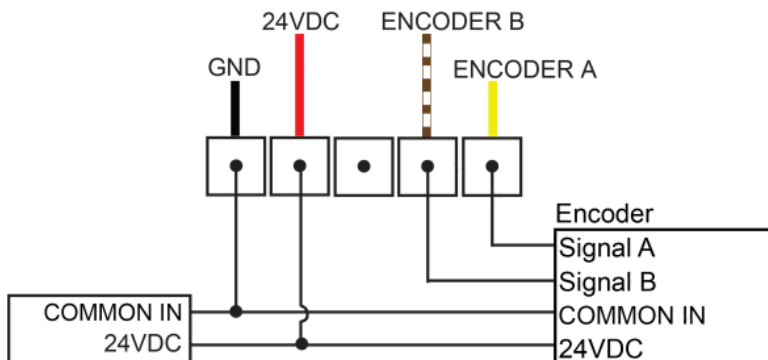
The In-Sight 3800L has two hardware encoder lines. These support a single channel (one input) and quadrature (two inputs) options. Depending on the encoder type, the pulse inputs can be interpreted for half or quarter increments. This means that when you are adjusting the ticks/pulses per line setting, your minimum ticks will be determined based on the encoder type.



5V encoder signals are not supported. The supported encoder voltage range is 10-30 VDC. The maximum frequency of 500kHz is only supported with 15m long cables.



To use the encoder, connect the A and B signals to the breakout connections on Pins 1 (yellow) and 4 (white/brown) respectively. Below is a schematic for the connections.



Encoder

When using a hardware encoder, there are some key considerations to keep in mind.

1. Place the encoder in a position where good contact with the moving surface can be maintained and has the same speed as the object that is being imaged. When integrating with a conveyor, make sure that the belt you are measuring is the same one you are imaging on!
2. Keep wire length from the encoder to the camera as short as possible. Long cable runs will often experience noise and voltage drop which will impact its performance.
3. Many encoders now have the ability to change the output pulses per revolution (ppr). Make sure that this corresponds to the ppr you are expecting on the camera's side.

Software Encoder

The software encoder tells the acquisition to take place at a consistent line rate for a certain time period in microseconds. This is helpful when you are not able to install a physical encoder, or you are in the testing phase and no movement of the system can happen yet.

When using the software encoder on the In-Sight 3800L, you can set the time period within the acquisition settings anywhere from 1 μ s – 1000 μ s. This parameter can be changed between acquisitions if needed.

Key aspects to keep in mind when using the software encoder.

1. If the speed of the system changes, even slightly, this will cause distortions in your image.
2. If using the software encoder for an initial setup/alignment of the system, remember that new lines are created constantly and may appear wavy when making changes.
3. Using the software encoder is a helpful tool when determining if a hardware encoder is causing an issue and for systems that have a relatively smooth and consistent speed, they can act as a temporary fix until the encoder can be replaced/fixed.

Squaring the pixels

Since line scans build the image slice by slice, the rate that the slices come in can vary unlike a traditional area scan camera. This can cause an image to look stretched or shrunk resulting in an aspect ratio that is not correct.



This is affected by the ppr of the hardware encoder or the time interval of the software encoder. As an example for the next two sections, we will use the pixel size that we calculated from the lensing exercise and pair it with a system that needs to operate at 650 ft/min.

Hardware Encoder

When using a hardware encoder, the pulses per rev (ppr) parameter is a determining factor in how fine of a resolution you can achieve as well as the rate that you can acquire at. This means that you can get a pulse out for a very defined distance and a repeatable rate.

Encoder

Typically, these encoders will have a wheel around them that contacts the movement surface. When the surface does not allow for a directly applied wheel, many times the encoder can be coupled directly with the rolling shaft that drives the movement. For either case, the circumference of the wheel or shaft diameter making contact needs to be known to get the proper distance per encoder tick/pulse.

$$\frac{Distance}{Tick} = \frac{circumference}{pulses\ per\ rev}$$

For example, if we have an encoder with a wheel that has a 12 in. circumference and a ppr of 12, we will get 1 pulse for every 1 inch of linear travel.

$$\frac{Distance}{Tick} = \frac{12\ in}{12\ ppr} = 1\ \frac{in}{tick}$$

This formula is what we will use when calculating the resolution needed to square the pixels in our previous example.

Based on our example from the lens determination section, we want our pixels to be 0.052mm wide and squared off. This means that we want to be able to get down to at least 1 pulse per 0.052mm of travel of the system. Given the previous example, a low ppr will not be sufficient in this example. We can re-arrange the above formula to give us the minimum ppr needed to obtain at least one tick per 0.052mm of travel.

$$Pulses\ Per\ Rev_{Minimum} = \frac{circumference}{target\ distance\ per\ tick} = \frac{12\ in}{0.052mm} \left(\frac{25.4mm}{in} \right) \approx 5862\ ppr$$

This outcome specifies the minimum encoder ticks needed to get *at least* 1 tick per 0.052 allow us to take an acquisition at the correct rate to square our pixels. However, most encoders do not have the ability to reach this resolution and will come with pre-defined selectable or non-selectable ppr. So, this is where the option to specify the ticks/steps per line will be helpful. This allows us to specify how many ticks generated correspond to our desired capture rate.

$$Ticks\ Per\ Line = \frac{PPR_{Encoder}}{circumference} \left(\frac{distance}{line} \right)$$

Let's use the example of a 16,000 ppr single channel encoder.

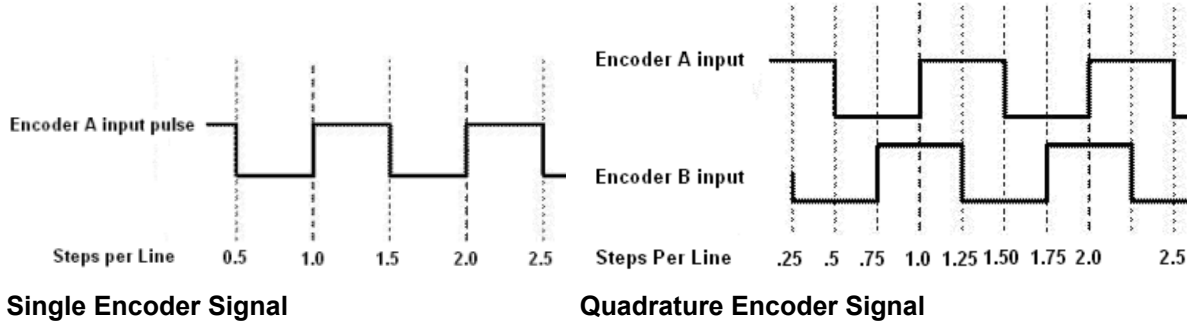
$$Ticks\ Per\ Line = \frac{16,000\ ticks}{12\ in} \left(\frac{0.052mm}{line} \right) \left(\frac{1in}{25.4mm} \right) \approx 2.730\ \frac{ticks}{line}$$

This example considers that each tick is only for the rising edge with a single channel encoder. This means that each tick is equivalent to 1 pulse. Said differently, the number of ticks per rev equals the number of pulses per rev. With quadrature encoder signals, some encoders will count every transition as a whole number. This is different from how In-Sight 3084L interprets the quadrature input signal.

The In-Sight 3800L counts the encoder's A channel rising edge transitions as whole numbers, but other transitions are detected as fractions. Therefore In-Sight quadrature signals are divided down by 4. For example: An encoder gives quadrature outputs capable of 16,000 cycles per full revolution. If In-Sight acquires 1 line per revolution, this value in steps per line for In-Sight 3800L is 4000 (or 16000 cycles divided 4) because In-Sight counts each transition as fractions. As a result, In-Sight will count each transition (1, 1.25, 1.50, 1.75, 2, 2.25, etc.) until the value of 4000 is reached. You receive the same resolution; it is interpreted differently in In-Sight.

Encoder

This is not the case with every encoder as some quadrature encoders will specify the ppr in rising edge cycles of the A-channel of the encoder. Therefore, it's important to know your specific encoder's definition for its ppr. Below are diagrams showing how the In-Sight system will interpret the encoder signals.



Given this condition, it's important to determine an effective ppr based on your encoder.

For a single channel encoder, this would be the following:

$$Effective\ PPR_{Rising\ only} = \frac{1\ pulse}{1\ step}$$

$$Effective\ PPR_{Rising\ and\ Falling} = \frac{2\ pulse}{1\ step}$$

For a quadrature encoder, this would be the following:

$$Effective\ PPR_{Rising\ only} = \frac{1\ pulse}{1\ step}$$

$$Effective\ PPR_{Rising\ and\ Falling} = \frac{4\ pulse}{1\ step}$$

When dealing with In-Sight Vision Suite, the goal is to get our encoder signals into a steps per line value. When trying to calculate the number of ticks needed to get to our goal of 0.052mm, we can use the following equations:

$$Steps\ Per\ Line_{Single} = \left(\frac{PPR_{Effective}}{circumference} \right) \left(\frac{distance}{line} \right) \text{ Rounded to 0.5 increments}$$

$$Steps\ Per\ Line_{Quadrature} = \left(\frac{PPR_{Effective}}{circumference} \right) \left(\frac{distance}{line} \right) \text{ Rounded to 0.25 increments}$$

If our encoder counts only the rising edges of the A-Channel for its definition of ppr, our new steps per line would be the following with 16,000 ppr:

$$Steps\ Per\ Line_{Single} = \left(\frac{16,000\ pulses}{12\ in} \right) \left(\frac{1\ step}{1\ pulse} \right) \left(\frac{0.052\ mm}{line} \right) \left(\frac{1\ in}{25.4\ mm} \right) \approx 2.730 \approx 3 \frac{steps}{line}$$

$$Steps\ Per\ Line_{Quadrature} = \left(\frac{16,000\ pulses}{12\ in} \right) \left(\frac{1\ step}{1\ pulse} \right) \left(\frac{0.052\ mm}{line} \right) \left(\frac{1\ in}{25.4\ mm} \right) \approx 2.730 \approx 2.75 \frac{steps}{line}$$

Encoder

For example, if the single channel encoder ppr is 16,000, we would get the following:

$$\begin{aligned} \text{Distance Per Line Final}_{\text{single}} &= \left(\frac{\text{steps}}{\text{line}} \right) \left(\frac{\text{Circumference}}{\text{ppr}_{\text{effective}}} \right) = \left(3.0 \frac{\text{steps}}{\text{line}} \right) \left(\frac{12 \text{ in}}{16,000 \text{ ppr}} \right) \left(\frac{25.4 \text{ mm}}{\text{in}} \right) \\ &\approx 0.057 \frac{\text{mm}}{\text{line}} \end{aligned}$$

If we use a quadrature encoder for the same solution, we would get:

$$\begin{aligned} \text{Distance Per Line Final}_{\text{Quadrature}} &= \left(\frac{\text{steps}}{\text{line}} \right) \left(\frac{\text{Circumference}}{\text{ppr}_{\text{effective}}} \right) = \left(2.75 \frac{\text{steps}}{\text{line}} \right) \left(\frac{12 \text{ in}}{16,000 \text{ ppr}} \right) \left(\frac{25.4 \text{ mm}}{\text{in}} \right) \\ &\approx 0.052 \frac{\text{mm}}{\text{line}} \end{aligned}$$

If our encoder counts both the rising edges AND falling edges for its definition of ppr, our new steps per line would be the following with 16,000 ppr:

$$\begin{aligned} \text{Steps Per Line}_{\text{single}} &= \left(\frac{16,000 \text{ pulses}}{12 \text{ in}} \right) \left(\frac{1 \text{ step}}{2 \text{ pulse}} \right) \left(\frac{0.052 \text{ mm}}{\text{line}} \right) \left(\frac{1 \text{ in}}{25.4 \text{ mm}} \right) \approx 1.36 \approx 1.5 \frac{\text{steps}}{\text{line}} \\ \text{Steps Per Line}_{\text{Quadrature}} &= \left(\frac{16,000 \text{ pulses}}{12 \text{ in}} \right) \left(\frac{1 \text{ step}}{4 \text{ pulse}} \right) \left(\frac{0.052 \text{ mm}}{\text{line}} \right) \left(\frac{1 \text{ in}}{25.4 \text{ mm}} \right) \approx 0.682 \approx 0.75 \frac{\text{steps}}{\text{line}} \end{aligned}$$

For example, the single channel encoder, we would get the following:

$$\begin{aligned} \text{Distance Per Line Final}_{\text{single}} &= \left(\frac{\text{steps}}{\text{line}} \right) \left(\frac{\text{Circumference}}{\text{ppr}_{\text{effective}}} \right) = \left(1.5 \frac{\text{steps}}{\text{line}} \right) \left(\frac{12 \text{ in}}{8,000 \text{ ppr}} \right) \left(\frac{25.4 \text{ mm}}{\text{in}} \right) \\ &\approx 0.057 \frac{\text{mm}}{\text{line}} \end{aligned}$$

If we use a quadrature encoder for the same solution, we would get the following:

$$\begin{aligned} \text{Distance Per Line Final}_{\text{Quadrature}} &= \left(\frac{\text{steps}}{\text{line}} \right) \left(\frac{\text{Circumference}}{\text{ppr}_{\text{effective}}} \right) = \left(0.75 \frac{\text{steps}}{\text{line}} \right) \left(\frac{12 \text{ in}}{4,000 \text{ ppr}} \right) \left(\frac{25.4 \text{ mm}}{\text{in}} \right) \\ &\approx 0.057 \frac{\text{mm}}{\text{line}} \end{aligned}$$

You might still need to adjust in which case, you can increase the amount of ppr of the encoder or you can decrease the size of the attached wheel (if not shaft mounted) to obtain the resolution required.

Another consideration when choosing the correct ppr for an encoder is the frequency of the pulses coming in at speed. For the In-Sight 3800L, the max input frequency (Hz) in ticks per second is 500kHz.

$$\text{Encoder Frequency} = \frac{\text{speed} * \text{ppr}}{\text{circumference}}$$

For our given example, the speed we are trying to maintain is 650ft/min.

$$\text{Encoder Frequency} = \frac{\left(650 \frac{\text{ft}}{\text{min}} \right) 16,000 \text{ ppr}}{12 \text{ in}} \left(\frac{12 \text{ in}}{1 \text{ ft}} \right) \left(\frac{1 \text{ min}}{60 \text{ s}} \right) \approx 173.3 \text{ kHz}$$

Given that this is well below the frequency of the max frequency of the In-Sight 3800L, we know that at this line speed, we will not overrun the input to the camera.

Encoder

Software Encoder

The software encoder sets the time between acquired lines known as the line period. When using this, the actual speed of the object is not considered, and the acquisition will always take place at the specified rate. This will result in stretching/shrinking of the object in the field of view during ramp up and ramp down periods of movement. When using the software encoder, you are assuming that your speed is very consistent and that any ramp up/down periods take place when the object is not being imaged.

With these assumptions, we can again work to square the pixels by determining the line period to acquire the desired 0.052mm pixel size in the vertical direction. This is usually calculated to an interval in μs . Doing so we can use the following formula:

$$\text{Line Period} = \frac{\text{distance per line}}{\text{speed}}$$

Plugging in our parameters from the previous example we get the following:

$$\text{Line Period}_{\mu s} = \frac{0.052mm \left(\frac{1 \text{ in}}{25.4 \text{ mm}} \right) \left(\frac{1 \text{ ft}}{12 \text{ in}} \right)}{\left(650 \frac{\text{ft}}{\text{min}} \right) \left(\frac{1 \text{ min}}{60 \text{ s}} \right) \left(\frac{1 \text{ s}}{1000 \text{ ms}} \right) \left(\frac{1 \text{ ms}}{1000 \mu s} \right)} \approx 15.74 \mu s$$

We can convert this into a frequency by taking the reciprocal:

$$\text{Line Period}_{Hz} = \frac{1}{\text{line period}_{\mu s}} \left(\frac{1000 \text{ ms}}{1 \mu s} \right) \left(\frac{1000 \text{ s}}{1 \text{ ms}} \right) = 63.5 \text{ kHz}$$

The In-Sight 3800L has a max line rate with exposure time at 4k resolution of 84kHz (11.9 μs). Given that we have 63.5 kHz (15.74 μs) at our current rate, we are in a safe range.

Exposure

In the previous examples, we calculated the rate that we can acquire a new line. In those calculations, exposure time was not factored in. Line-scan exposure times are very small compared to area-scan systems. This is one of the reasons why the intensity and narrow angle of the bar lights used with them are very critical. There are two things to consider when choosing the exposure time:

1. The line period at max speed.
2. The speed it takes to move the equivalent of 1 pixel.

In the last example, we calculated that 15.74 μs (63.5 kHz) was our line rate for our speed of 650ft/min. Our max line acquisition rate is 84 kHz (11.9 μs). This 11.9 μs minimum is composed of exposure time and the time needed to read the signal from the sensor before starting the next acquisition.

The following recommendation for the exposure time of the is the following.

$$\text{Exposure}_{max} = \text{line period} - 0.5 \mu s$$

This is meant to ensure that the amount of exposure taken does not exceed your acquisition rate, resulting in dropped acquisitions. This is the maximum time you can set the exposure and typically the lower the exposure you can work with the better. The In-Sight 3800L has a minimum exposure time of 0.1 μs .

When you find that exposure time available is not enough, look into ways of either increasing your light's intensity if possible or seeing if your lens has a variable aperture that can be opened up a bit more to allow more light. If the aperture is opened more, you will decrease your depth of field. This is usually not an issue when the object's distance from the lens is constant, and a narrower depth of field is not as critical.

Encoder

Another method to keep in mind is if the spatial resolution is not needed and can be decreased, the In-Sight 3800L can use the 1K mode which is more sensitive to light increasing your overall brightness of the image. This will increase the effective pixel size allowing each pixel to receive more light per acquisition compared to the 2K and 4K modes.

Speed Considerations

The sections above have worked at a speed of 650 ft/min. While this is relatively high speed, there are situations where you will need to go faster. Below are some tips to handle situations where you need to optimize for higher speeds.

1. Consider your smallest pixel sizing needed to conduct the inspection. In many cases, you do not need to use the full resolution capabilities of the In-Sight 3800L and can select a smaller resolution such as the 2K mode. In this mode, your max line rate is about double that of the 4K and 1K modes.
2. Consider your necessary image size. Sometimes you do not need the max length allowable (16,376 lines) and trimming this up can help reduce your processing/networking overhead. Consider what (if any) overlapping you really need. If your experienced defects are small relative to your overlap, you could decrease this, and the overall image size needed.
3. Decrease your exposure as far as possible and increase (if possible) your lighting intensity. This will allow you to possibly increase your line rate (remember that there is a minimum threshold depending on what resolution mode you are in) and decrease the amount of blur within your image. This is advantageous especially when using the 2K mode as it has the highest max line rate of 169 kHz.
4. Crop down your field of view in the horizontal direction. If you are not using the full width of your field of view, consider cropping how wide your image is to decrease the overall size of your image to be processed. When using image processing tools, also evaluate the region of interests (ROIs) to see if they are sized appropriately.

Troubleshooting Line Scan

11 Troubleshooting Line Scan

Below are some typical issues you might encounter when using a line-scan system such as the ISIn-Sight 3800L.

Description	Acquired Image
Correct Image	
Scrolling/Moving part in FOV	
Stretched	
Compressed	
Repeated Part in single image FOV	
Truncated	

Troubleshooting Line Scan

Below are some ways to handle the situations shown above.

Scrolling

Image scrolling is a result of an acquisition trigger problem. The trigger is received at inconsistent times, so the start of the image frame is varied. To stop image scrolling, make sure the image trigger is occurring at the correct expected time.

Stretched

A stretched image is a result of too many lines in the image. This is caused by the line rate being too **fast** for the movement of the conveyer belt. Adjust the line rate (decrease line period or steps per line depending on the encoder type) to compensate. Run the line period or encoder ticks per line calculations to see if your rate is appropriate. Typically, a simple increase in either Line Period or steps per line adjustment can be made to compensate for this.

Compressed

A compressed image is a result of too few lines in the image. This is caused by the line rate being too **slow** for the movement of the conveyer belt. Adjust the line rate (increase Line Period or Steps per line depending on the encoder type) to compensate. Run the line period or encoder steps per line calculations to see if your rate is appropriate. Typically, a simple increase in either Line Period or steps per line adjustment can be made to compensate for this.

Repeated Part

Repeated part in a single image is a result of an acquisition trigger is not being received when expected or building the image with too many lines. Adjust the acquisition trigger or the total number of lines to acquire.

Truncated

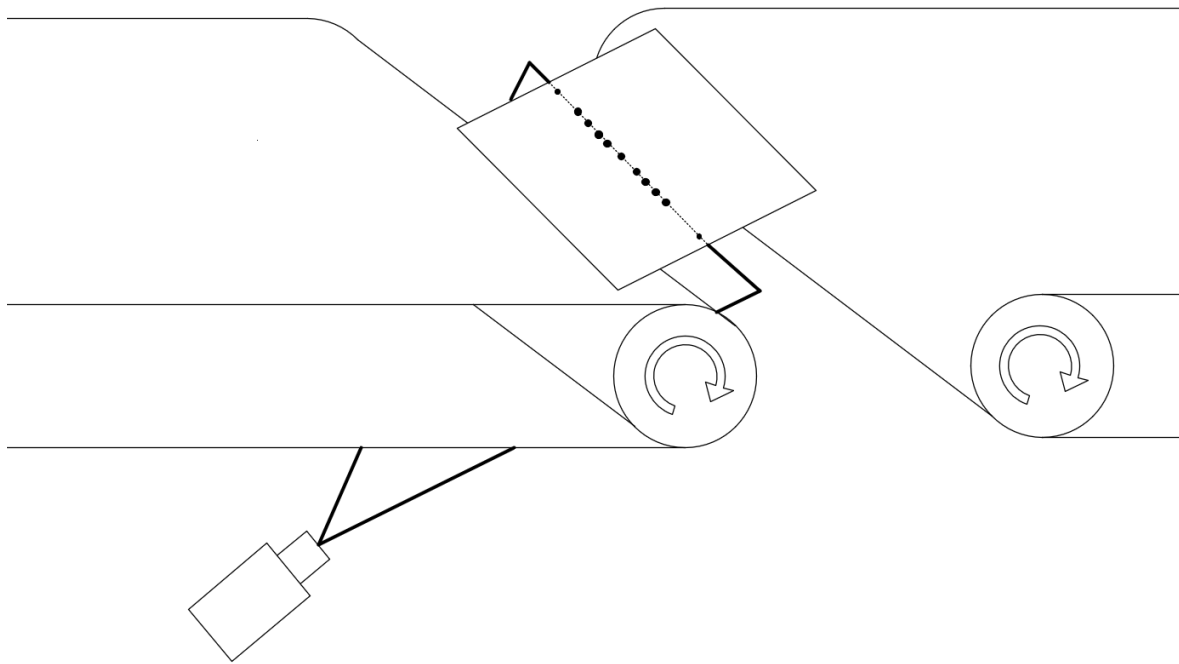
A truncated part is a result of the number of lines for the total image is too small or an unexpected acquisition trigger was received. Adjust the acquisition trigger or the number of total lines in the image to compensate. A truncated image could be seen when Line Clip mode is set to Reduced Line Height. If this is the case, the truncated image is expected.

Appendix A System Alignment Techniques

Appendix A System Alignment Techniques

One of the trickiest parts when working with a line-scan system is getting the alignment completed correctly. Unlike a traditional area scan, the line-scan system will only see one row of pixels at a time, which makes it very difficult to know what you are looking at within the field of view. Thankfully, this is a relatively simple method to help get the alignment close quickly and allow for final adjustments.

Let's assume that you are looking to position a line scan to look at a particular gap within a conveyor belt system so you can see the underside of packages as they go across it. Using the sheet in the next page or something similar, place that on the top side of the belt gap so that the series of dots correspond with the line that you want to acquire. The diagram below shows the line-scan system placed below the belt gap with the sample target sheet below. Note that the dots are shown on the top side of the page in this diagram to better show how to align. However, when doing this on a real system, you want the dots facing downwards towards the camera lens. The dotted line is meant to show the representation of the FOV width aligning to the dots on the target page.

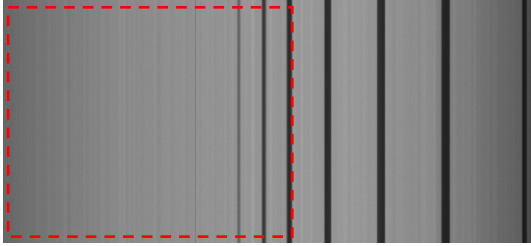


Next, set your line-scan system to use a software encoder with moderately high exposure time. As you trigger the camera, check your corresponding image.

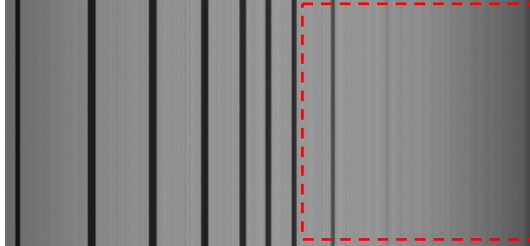
While checking your image previews, start to see if you can get a series of vertical bars to appear in your image. These vertical bars will represent the horizontal positions of the dots that are printed on your alignment sheet.

Appendix A System Alignment Techniques

If you get most of the correct number of bars on the right side but not the left side, twist your positioning counterclockwise slightly until you start to see the correct number of vertical lines. Likewise, if you see the correct number of vertical lines on the left but not the right, twist your system positioning clockwise slightly until you get the correct number of vertical lines.

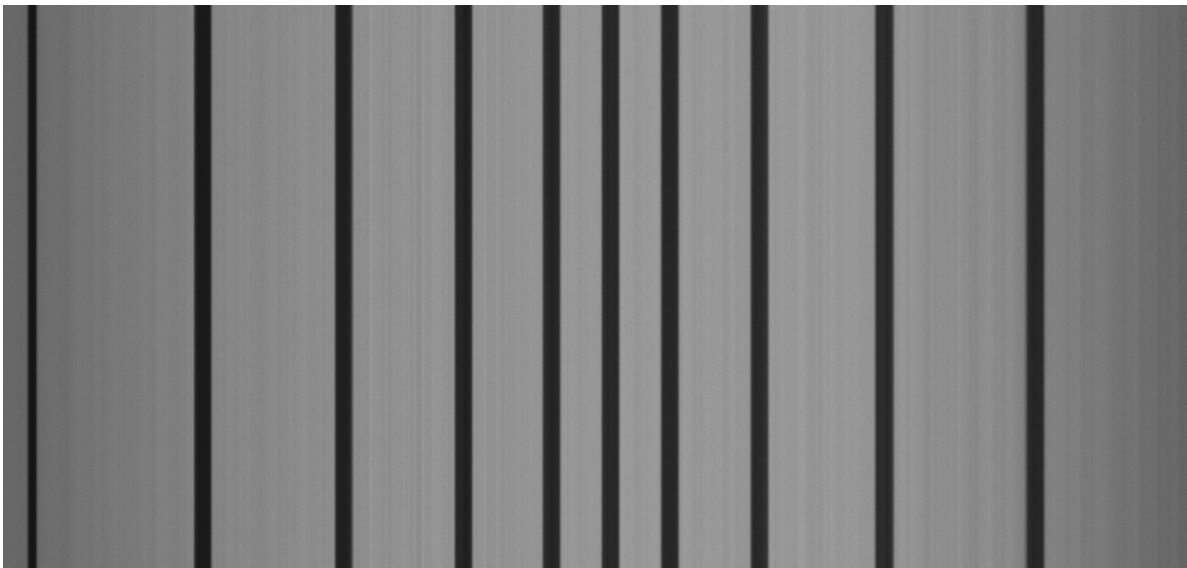


Most bars on the right but not left. Twist the alignment of the camera counterclockwise



Most bars on the right but missing on the left side of the image. Twist the alignment of the camera clockwise.

The goal is to get all vertical lines to be visible with consistent widths and distances. Below is an example of proper alignment of the target sheet.



All bars displayed and in the correct alignment

Once you have the correct number of vertical lines, you can check the system by running an actual part and checking the skew. This method of alignment also works when adjusting your lighting. This is much easier to see take place. Taking the light's narrow beam, align it with the dots on the page as you did with the line scan alignment. Your resulting lighting will appear even across the image once properly aligned.

If your image has the correct alignment and the image is evenly illuminated, secure your system in place so that any focus adjustments that need to be made do not affect your positioning.

Appendix A System Alignment Techniques

•

•

•

•

•

•

•

•

•

•

•

Appendix A System Alignment Techniques

Formulas

Formula	Description
$Focal\ Length_{width} = \frac{Working\ Distance * Sensor_{width}}{Field\ of\ View_{width} + Sensor_{width}}$	Use this to determine the focal length of the camera needed at the desired working distance. Sensor _{width} is the physical size of the pixel. All units need to be the same.
$Resolution_{mm/px} = \frac{field\ of\ view_{width}mm}{sensor\ pixel\ count\ px}$	Use this to determine the spatial resolution of your image. Note that this will give you the target size to set your encoder settings to maintain square pixels.
$Field\ Of\ View_{Length} = PixelSize * Number\ Of\ Acquired\ Lines$	This is used to determine your field of view in the Y-direction or length of your final image.
$Trigger\ Rate_{ms} = \frac{(full\ image\ length_{px} - overlap_{px})}{* line\ period_{ms}}$	Set this to maintain your overlap size for a set full image size.
$\frac{Distance}{Tick} = \frac{circumference}{pulses\ per\ rev}$	This will give your encoder nominal resolution in distance/tick.
$Pulses\ Per\ Rev_{Minimum} = \frac{circumference}{target\ distance\ per\ tick}$	Use this to determine the minimum encoder resolution needed to maintain your square pixels.
$Effective\ PPR_{Rising\ only} = \frac{1\ pulse}{1\ step}$ $Effective\ PPR_{Rising\ and\ Falling} = \frac{4\ pulse}{1\ step}$	Use this to determine the effective ppr seen by In-Sight based on if your encoder's ppr is counted as only rising edge only or if rising AND falling edges are counted and if the encoder is a quadrature type.
$Effective\ PPR_{Rising\ only} = \frac{1\ pulse}{1\ step}$ $Effective\ PPR_{Rising\ and\ Falling} = \frac{2\ pulse}{1\ step}$	Use this to determine the effective ppr seen by In-Sight based on if your encoder's ppr is counted as only rising edge only or if rising AND falling edges are counted and if the encoder is a single channel type.

Appendix A System Alignment Techniques

$\text{Steps Per Line}_{\text{Single}} = \left(\frac{\text{PPR}_{\text{effective}}}{\text{circumference}} \right) \left(\frac{\text{distance}}{\text{line}} \right) \text{ Rounded to 0.5 increments}$ $\text{Steps Per Line}_{\text{Quadrature}} = \left(\frac{\text{PPR}_{\text{effective}}}{\text{circumference}} \right) \left(\frac{\text{distance}}{\text{line}} \right) \text{ Rounded to 0.25 increments}$	Use this to determine how many encoder ticks/pulses are required before the camera starts a new acquisition based on single channel or quadrature encoder signals with an effective ppr and $\text{PPR}_{\text{encoder}} > \text{PPR}_{\text{minimum}}$.
$\text{Distance Per Line}_{\text{final}} = \frac{\text{ticks}}{\text{line}} \left(\frac{\text{Circumference}}{\text{ppr}} \right)$	This will output your final Y-direction resolution based on your final Ticks/Line rate.
$\text{Encoder Frequency} = \frac{\text{speed} * \text{ppr}}{\text{circumference}}$	Use this to check your encoder frequency.
$\text{Line Period} = \frac{\text{distance per line}}{\text{speed}}$	Use this to determine the time between acquisitions when using a software encoder.
$\text{Line Period}_{\text{Hz}} = \frac{1}{\text{line period}_{\mu\text{s}}} \left(\frac{1000 \text{ ms}}{1 \mu\text{s}} \right) \left(\frac{1000 \text{ s}}{1 \text{ ms}} \right)$	Use this to convert your line period (in μs) to frequency.
$\text{Exposure}_{\text{max}} = \text{line period} - 0.5 \mu\text{s}$	Use this to determine the max exposure time possible without missing acquisitions.