GETTING STARTED WITH EASY RANGER

GBC08 – OEM Business Team

November 2018

GETTING STARTED WITH EASY RANGER

OVERVIEW OF PRESENTATION

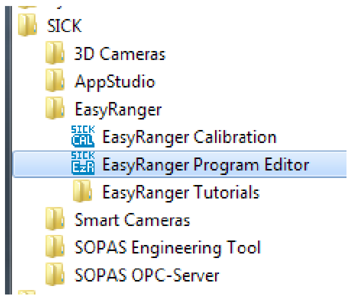
- General Introduction
  - How to get a license!
  - Introduction to Easy Ranger Program Editor
  - How to write programs in the Program Editor
  - Tips & Tricks in the Program Editor
- Tutorial 1
  - Detailed go-through of how to solve an application in Easy Ranger!

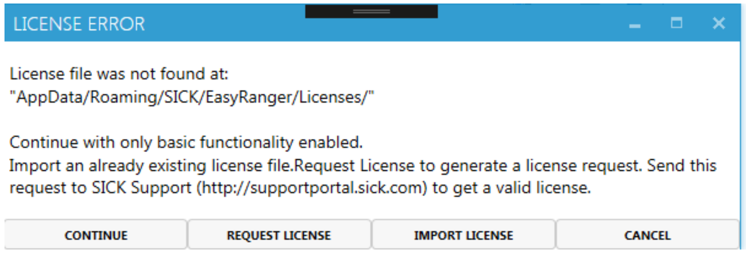
LICENSE

- The Easy Ranger installation contains two programs:
  - Easy Ranger Program Editor
  - Ranger3Setup
- To use the Program Editor you will need to have a license. There are a number of different licenses available:
  - **No license**: The program editor can only be used for loading and viewing images and other variables, no image processing or other analysis is available.
  - **Trial license**: A free time limited license for testing the program editor, can be used on multiple computers
  - **Developer license**: Premium license for developing applications in the program editor, is only valid on one computer.
  - **Run time license**: Can be used in production, applications can be executed but no further developent is possible.

LICENSE

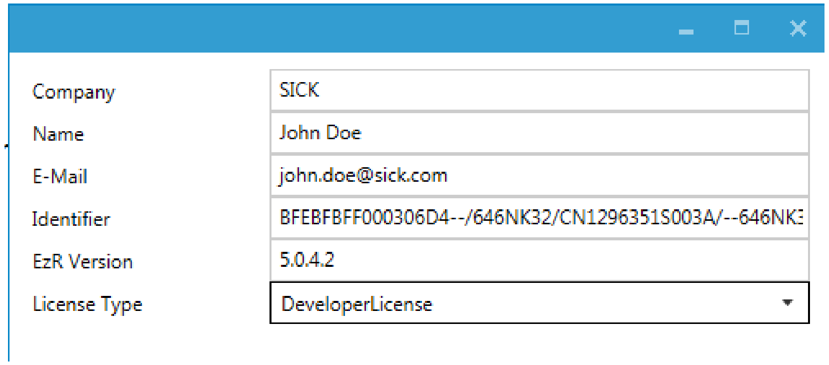
- To get a license follow the steps below:
  - Start the Program Editor from the Windows Start Menu
    - You will see a message box with a license error message
  - Press **Request License**
    - Fill in the form and click **Save to File**
    - Send the request file to SICK to obtain your license
- Start the program editor again and click **Import License**
- **Request License** and **Import License** can also be found in the help menu in the program editor




LICENSE ERROR

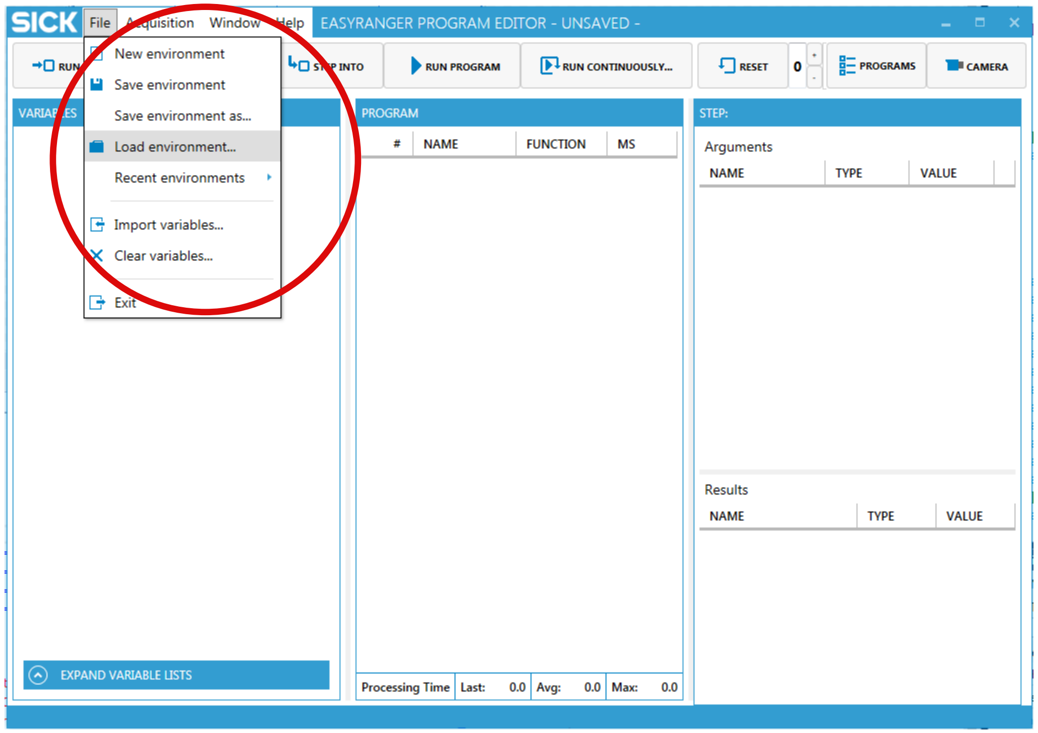
License file was not found at:
"AppData/Roaming/SICK/EasyRanger/Licenses/"

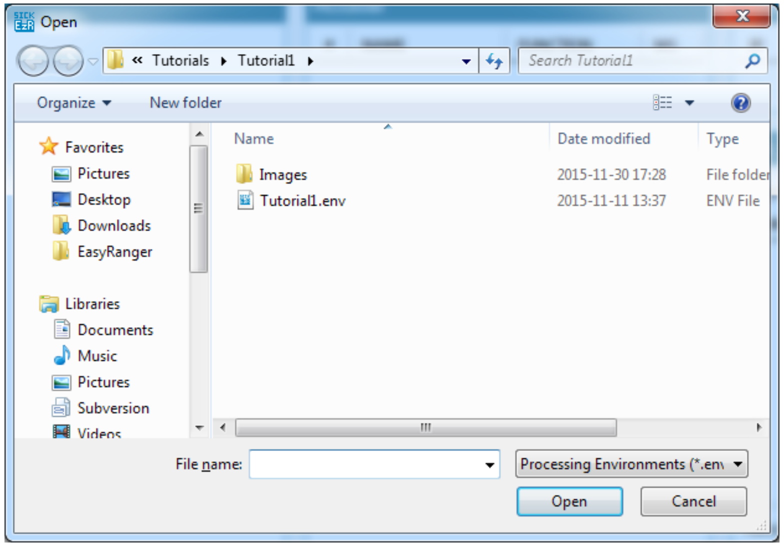
Continue with only basic functionality enabled.
Import an already existing license file.Request License to generate a license request. Send this request to SICK Support (http://supportportal.sick.com) to get a valid license.


Company	SICK
Name	John Doe
E-Mail	john.doe@sick.com
Identifier	BFEBFBFF000306D4--/646NK32/CN1296351S003A/--646NK3
EzR Version	5.0.4.2
License Type	DeveloperLicense

GENERAL INTRODUCTION
THE FIRST STEPS…

- Start the Program Editor
  - Load the Tutorial1.env configuration
  - We will use this to describe how to work with the Easy Ranger Program Editor





Name	Date modified	Type
Images	2015-11-30 17:28	File folder
Tutorial1.env	2015-11-11 13:37	ENV File

The environment file (.env) contains both the program and the variables shown in the Program Editor

GENERAL INTRODUCTION
THE PROGRAM EDITOR UI

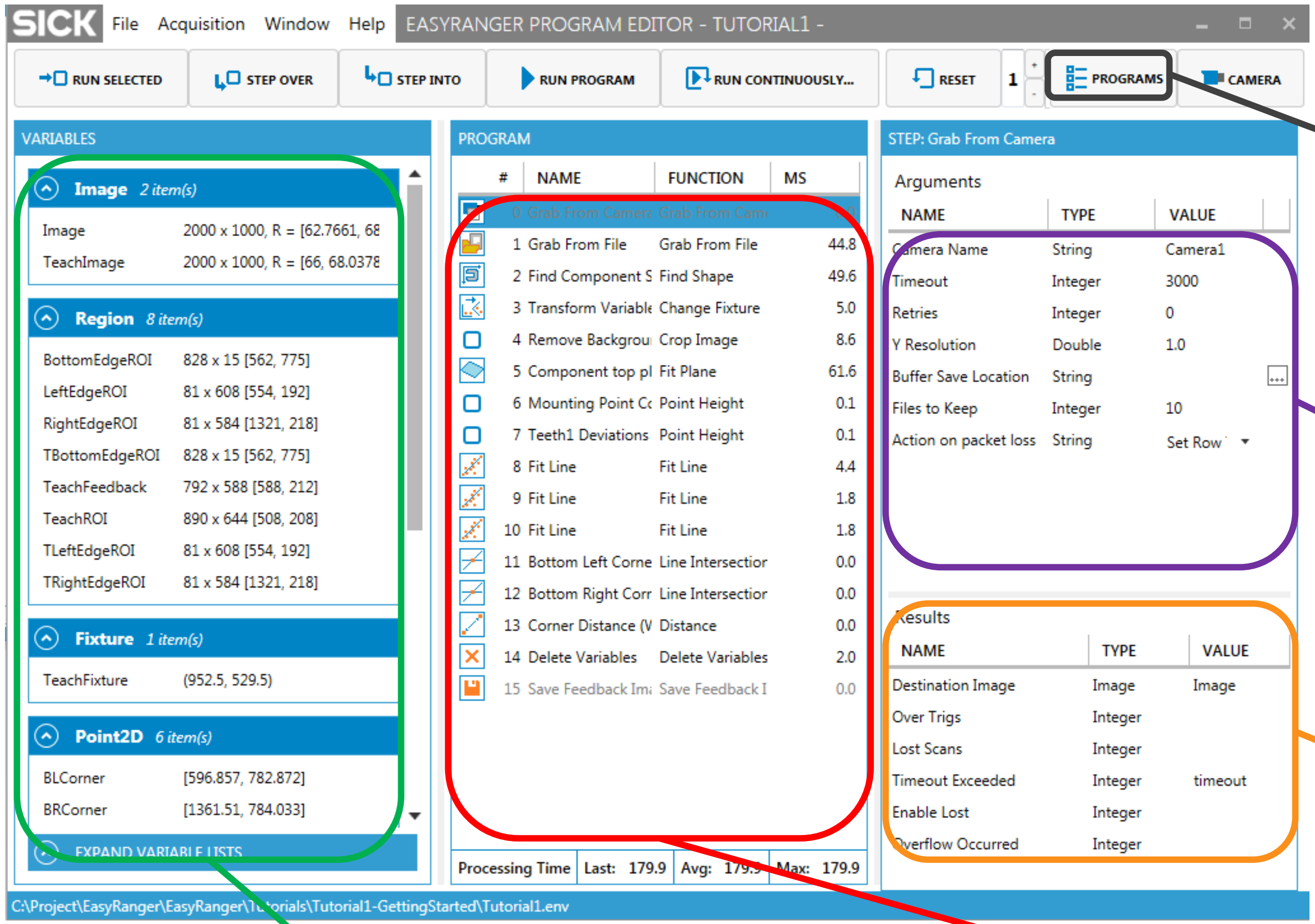


Image *2 item(s)*

Image	2000 x 1000, R = [62.7661, 68
TeachImage	2000 x 1000, R = [66, 68.0378

Region *8 item(s)*

BottomEdgeROI	828 x 15 [562, 775]
LeftEdgeROI	81 x 608 [554, 192]
RightEdgeROI	81 x 584 [1321, 218]
TBottomEdgeROI	828 x 15 [562, 775]
TeachFeedback	792 x 588 [588, 212]
TeachROI	890 x 644 [508, 208]
TLeftEdgeROI	81 x 608 [554, 192]
TRightEdgeROI	81 x 584 [1321, 218]

Fixture *1 item(s)*

TeachFixture	(952.5, 529.5)

Point2D *6 item(s)*

BLCorner	[596.857, 782.872]
BRCorner	[1361.51, 784.033]

#	NAME	FUNCTION	MS
0	Grab From Camera	Grab From Cam	
1	Grab From File	Grab From File	44.8
2	Find Component S	Find Shape	49.6
3	Transform Variable	Change Fixture	5.0
4	Remove Backgrou	Crop Image	8.6
5	Component top pl	Fit Plane	61.6
6	Mounting Point Cc	Point Height	0.1
7	Teeth1 Deviations	Point Height	0.1
8	Fit Line	Fit Line	4.4
9	Fit Line	Fit Line	1.8
10	Fit Line	Fit Line	1.8
11	Bottom Left Corne	Line Intersectior	0.0
12	Bottom Right Corr	Line Intersectior	0.0
13	Corner Distance (V	Distance	0.0
14	Delete Variables	Delete Variables	2.0
15	Save Feedback Ima	Save Feedback I	0.0

Processing Time | Last: 179.9 | Avg: 179.9 | Max: 179.9

Arguments

NAME	TYPE	VALUE
Camera Name	String	Camera1
Timeout	Integer	3000
Retries	Integer	0
Y Resolution	Double	1.0
Buffer Save Location	String	
Files to Keep	Integer	10
Action on packet loss	String	Set Row

Results

NAME	TYPE	VALUE
Destination Image	Image	Image
Over Trigs	Integer	
Lost Scans	Integer	
Timeout Exceeded	Integer	timeout
Enable Lost	Integer	
Overflow Occurred	Integer	

C:\Project\EasyRanger\EasyRanger\Tutorials\Tutorial1-GettingStarted\Tutorial1.env

Program selector

Selects the current step program

Step Arguments

Input to the currently selected step

Step Results

Variables where the step will store its outputs

Variable View

All the currently defined user variables; images, regions, points, lines, shapes, etc

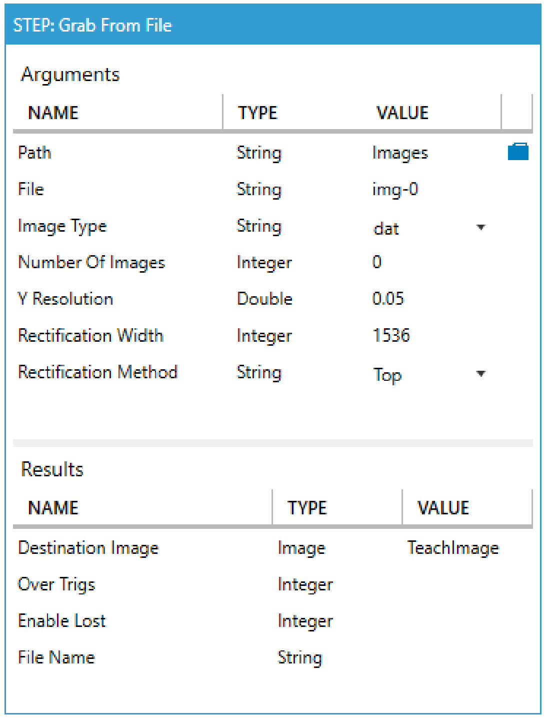
Step Program View

The sequence of tools used in the current step program. Executes from top to bottom, step-by-step, when you run the program

GENERAL INTRODUCTION
THE STEP PROGRAM VIEW

- Add tools by right-clicking in the program list
- Right click and select ”Show Description” to view the documentation of the tool (or press F1)
- Each tool has a set of arguments and results
  - Left click on the tool to see its current settings

- Select the **Teach Program** at the top of the Editor
  - This is done in the Program Selector bar
  - The first tool in the Teach program is **Grab From File**
  - We use it to acquire an image from disk and will explain it in detail later
  - The argument **Path** is set to ’Images’. The path is relative to the localisation of the environment file
  - The argument **File** is set to ’img-0’ which is the name of the teach image
  - For explanation about the other arguments, see the online documentation (F1)


STEP: Grab From File

Arguments

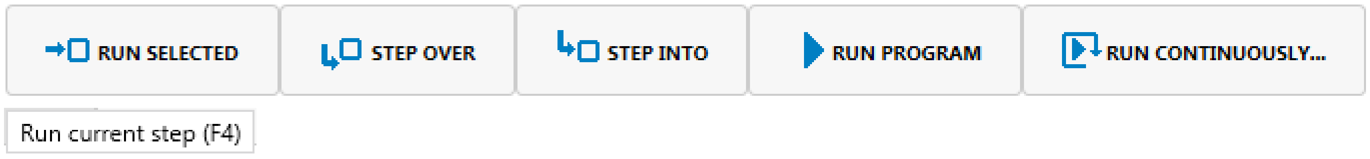
NAME	TYPE	VALUE
Path	String	Images
File	String	img-0
Image Type	String	dat
Number Of Images	Integer	0
Y Resolution	Double	0.05
Rectification Width	Integer	1536
Rectification Method	String	Top

Results

NAME	TYPE	VALUE
Destination Image	Image	TeachImage
Over Trigs	Integer	
Enable Lost	Integer	
File Name	String	

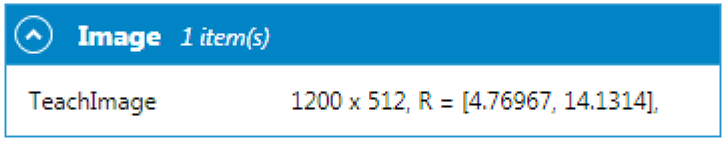
GENERAL INTRODUCTION
THE STEP PROGRAM VIEW CONTINUED..

- Select the first step (click on it to highlight it)
- Press the ’Run Selected’ button at the top of the window (F4-button)
  - The tool is now executed and the Image variable ’TeachImage’ is set accordingly.
  - See the shortcuts for the debug functions by hovering over the buttons
- All variables currently defined are seen on the left
  - The sorting is based on type
  - If the variable does not exist, the tool will create it
- Step program structure
  - Double-click to rename tool
  - Right-click for options
    (add step/tool, handle indentation, add comment etc..)



GENERAL INTRODUCTION
THE VARIABLE VIEW

- The following variable types are currently supported
  - Image, Region, Plane, Point2D, Point3D, Fixture, Line, Double, String, Text, ShapeLocator, Profile, TCPServer



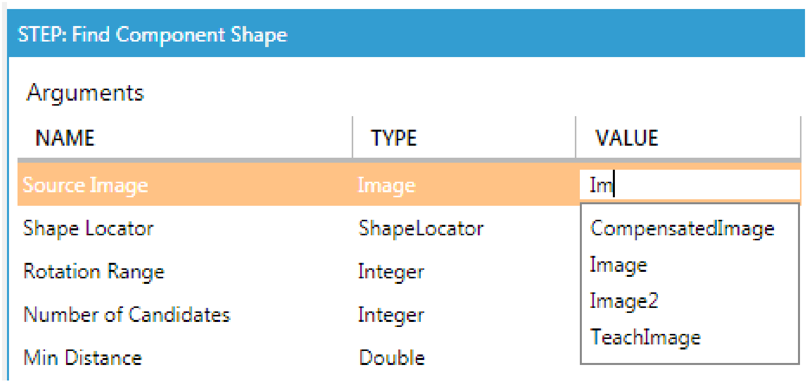
TeachImage 1200 x 512, R = [4.76967, 14.1314],

- Right-click for options
- Works as File Explorer, e.g. use Ctrl, Shift, arrow keys etc
- Double-click Image or Profile to open viewer
- For Points/Lines/Rectangles/Ellipses generated in the viewer
  - Right-click and select “Create Step” to create a Define Variable step with valid input arguments

GENERAL INTRODUCTION
THE STEP ARGUMENTS & RESULTS

SICK
Sensor Intelligence.

- For the variables there is an auto-completion function that lists all variables of the correct type
- Variables that have a predefined set of arguments are available as list items in a drop-down



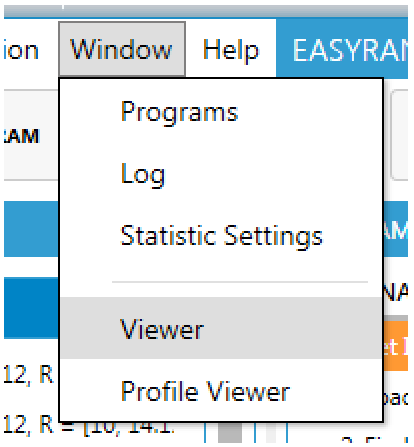
STEP: Find Component Shape

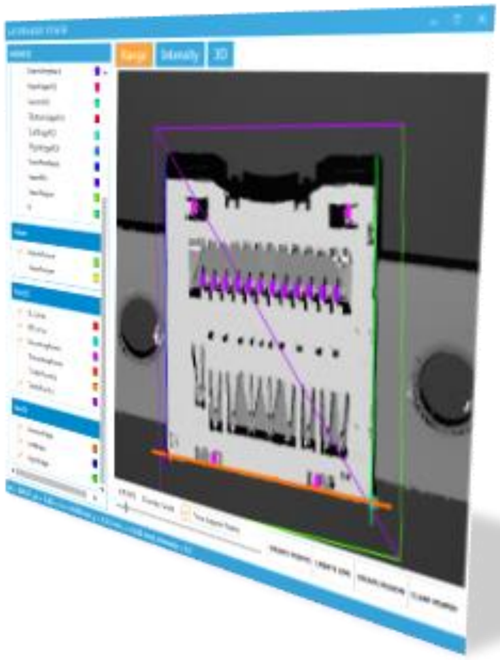
Arguments

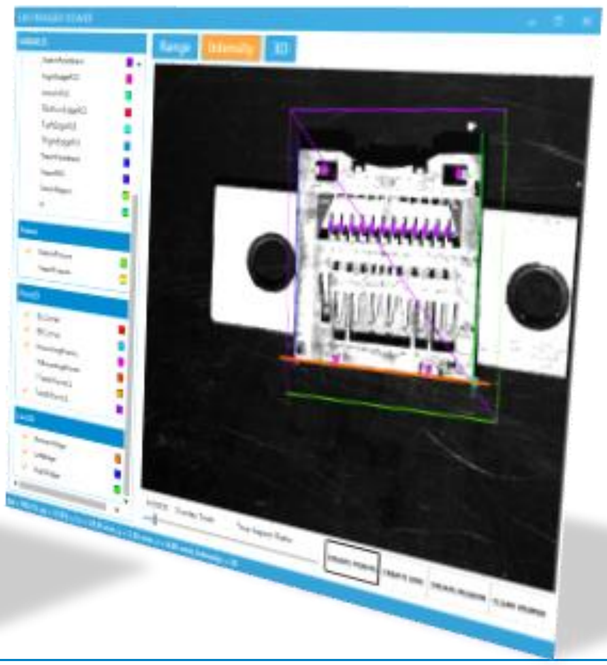
NAME	TYPE	VALUE
Source Image	Image	Im
Shape Locator	ShapeLocator	
Rotation Range	Integer	
Number of Candidates	Integer	
Min Distance	Double	

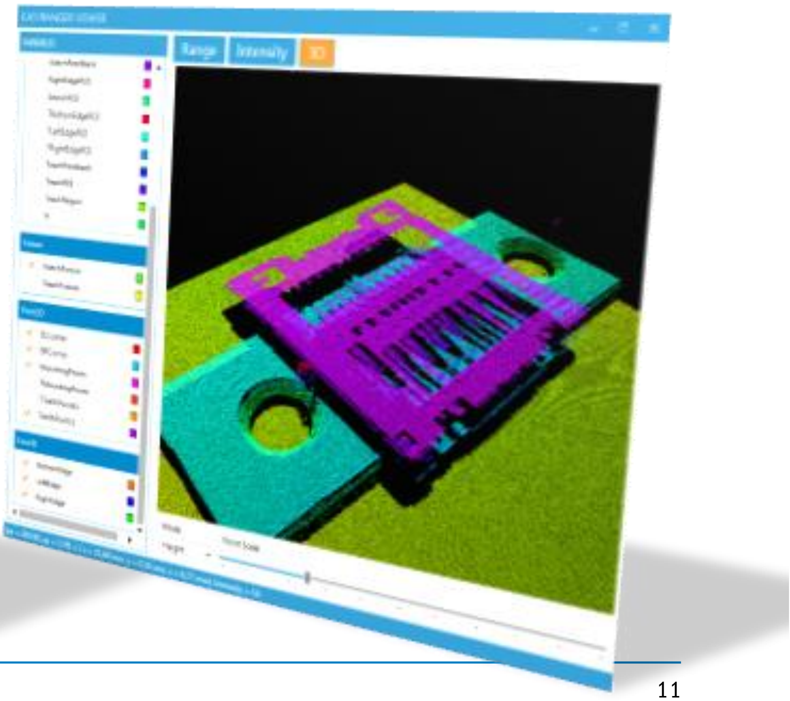
GENERAL INTRODUCTION
THE VIEWER

- To look at variables (images, points, lines, etc) you use the Easy Ranger Viewer
  - Double click on image or profile to open the viewer
  - You can select which variables to visualize by checking or unchecking the box to the left of the variable's name
  - You can create points, lines and regions graphically by pressing the buttons at the bottom right.



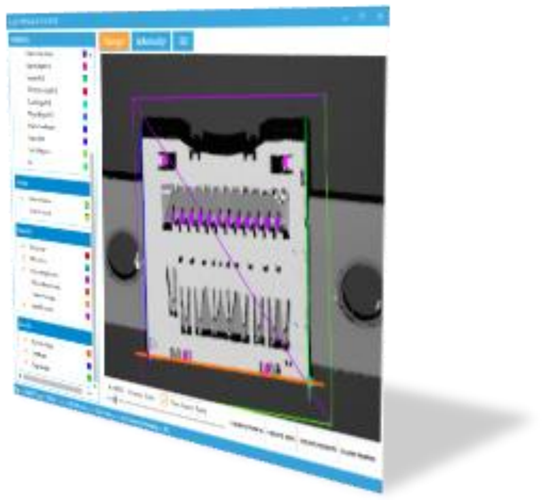




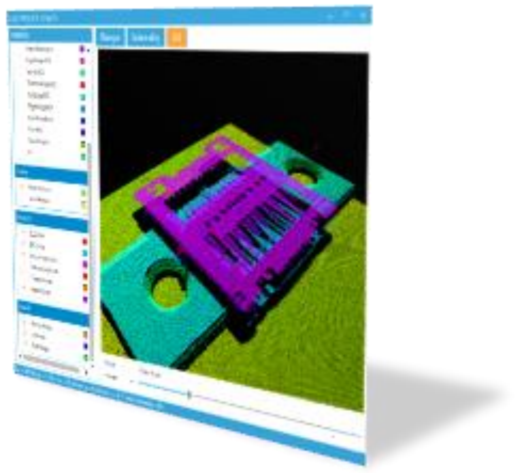


GENERAL INTRODUCTION
THE VIEWER - TIPS & TRICKS

- Handling Points/Lines/Rectangles/Ellipses (in Range & Intensity view)
  - Click on region to see position and dimension
  - Press Ctrl to move/resize/rotate elements (or use “Edit Variable” check box)
  - Press Ctrl + Shift to delete elements (or use “Delete Variable” check box)
  - When selected, click on individual elements to delete them



- Navigation in 3D view
  - Use scroll wheel/touch pad to zoom
  - Use left mouse button to rotate
  - Use right mouse button to pan
  - Use Shift + Left-mouse to heighten/lower the view (in relation to crosshair)



GENERAL INTRODUCTION
BUILDING UP THE SCRIPT...

- Just add the tools you need.
  - A tool can always use variables as input.
  - A **Fit Line** needs a region to work on. This region could be the result from e.g. a **Find Blobs** tool
  - The fitted line could e.g. be used to find a corner point. The **Line Intersection** tool uses two lines as input and returns a Point2D.
  - You can then use that point as input to e.g. a **Distance** tool to find the distance between a point and a line or two points.
  - Etc...

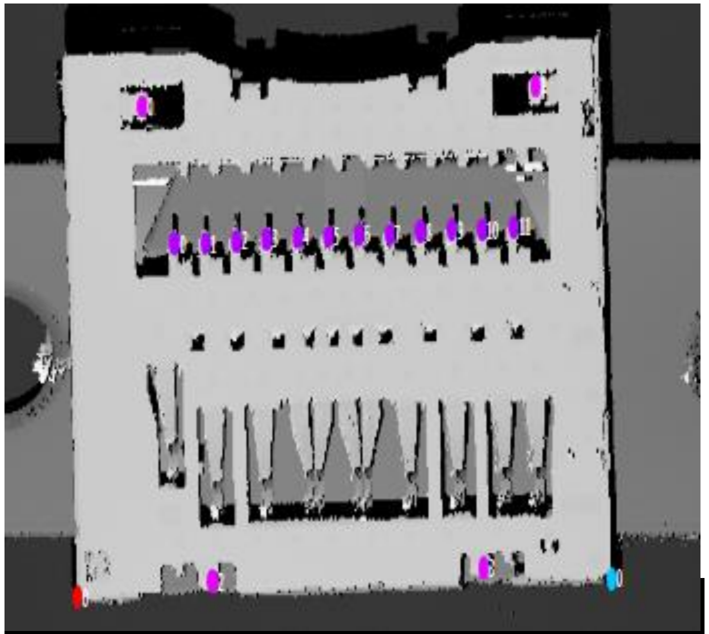
TUTORIAL 1

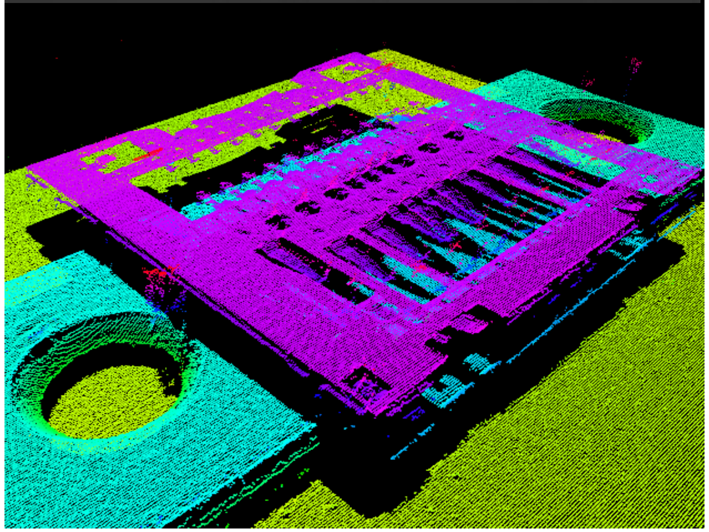
- On the following slides, the logical flow of the step program in Tutorial 1 is explained
  - For details on how a specific tool works, please consult the embedded manual.
    - Right click on a step and select **Show Description**
      - Or click F1 on your keyboard
    - Right click somewhere in the step program and select one of the **Add Step** options

TUTORIAL 1
APPLICATION DESCRIPTION

The tutorial program shows how to do the following:

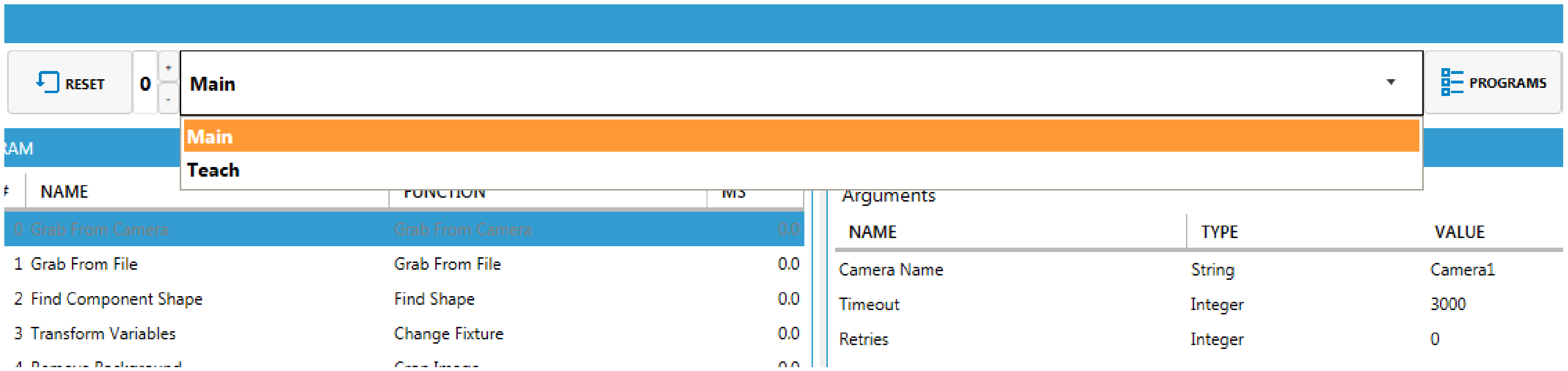
- Locate the connector component in the image
- Find the four mounting points
- Find the connector pins
- Measure co-planarity between mounting points and connector points
- Measure the distance from the two bottom corners





TUTORIAL 1
TEACH PROGRAM

- One environment can contain multiple programs
- This tutorial is split in two parts: Main & Teach
- Select the Teach program
  - Click Run Program (F5) to load the teach image, crop out the background and teach the shape of the component that will be used for localization
- Switch to the Main program, which we will go through in more detail

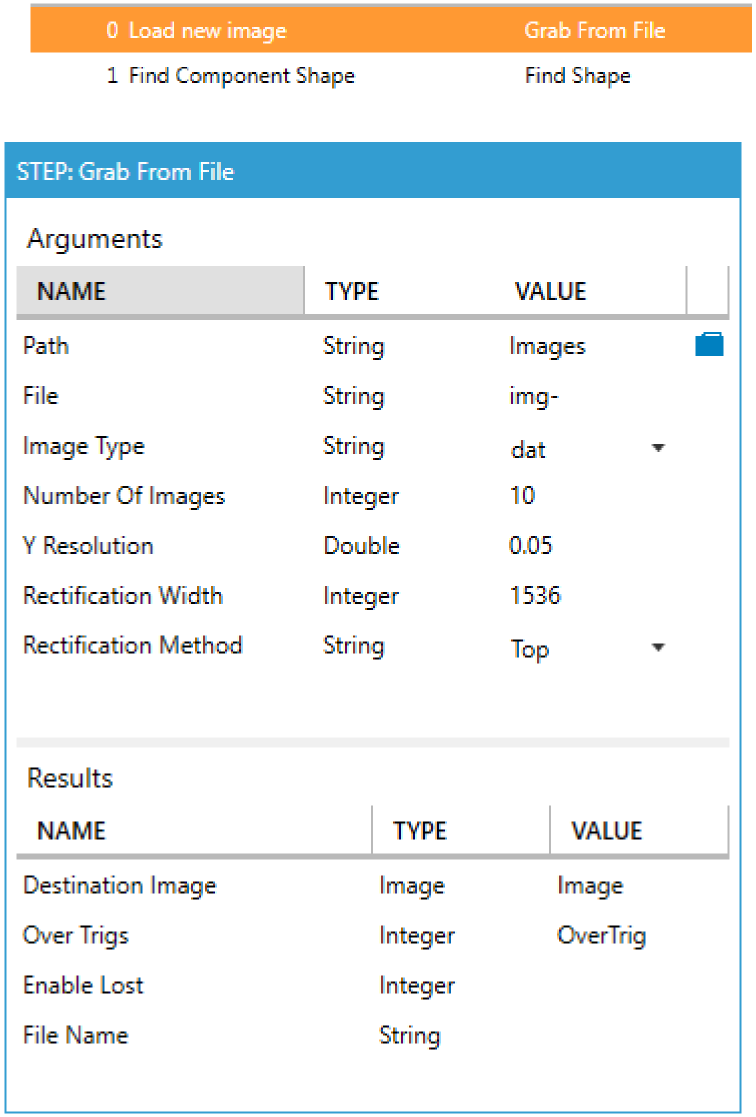


#	NAME	FUNCTION	MS
0	Grab From Camera	Grab From Camera	0.0
1	Grab From File	Grab From File	0.0
2	Find Component Shape	Find Shape	0.0
3	Transform Variables	Change Fixture	0.0
4	Remove Background	Crop Image	0.0

NAME	TYPE	VALUE
Camera Name	String	Camera1
Timeout	Integer	3000
Retries	Integer	0

TUTORIAL 1
LOADING IMAGES

- Data can be acquired from disk or directly from a camera
  - The **Grab From File** (Acquisition) tool loads iConBuffers from disk
  - In this case it loads the data from the subfolder *Images*. The filename is *img-* followed by a number which increases by 1 each time the step is executed.
  - Enable Counter = 10 means the number will wrap around at 10. This means the program will loop over 10 images indefinitely if you let it run.
  - The image buffers do not contain information about y resolution. It is therefore an argument to set the profile distance or encoder resolution if mark is available in the buffer.
- The resulting image will be stored in a variable named *Image*



0	Load new image	Grab From File
1	Find Component Shape	Find Shape

STEP: Grab From File

Arguments

NAME	TYPE	VALUE
Path	String	Images
File	String	img-
Image Type	String	dat
Number Of Images	Integer	10
Y Resolution	Double	0.05
Rectification Width	Integer	1536
Rectification Method	String	Top

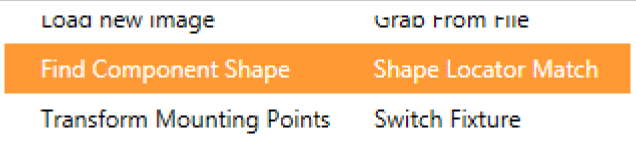
Results

NAME	TYPE	VALUE
Destination Image	Image	Image
Over Trigs	Integer	OverTrig
Enable Lost	Integer	
File Name	String	

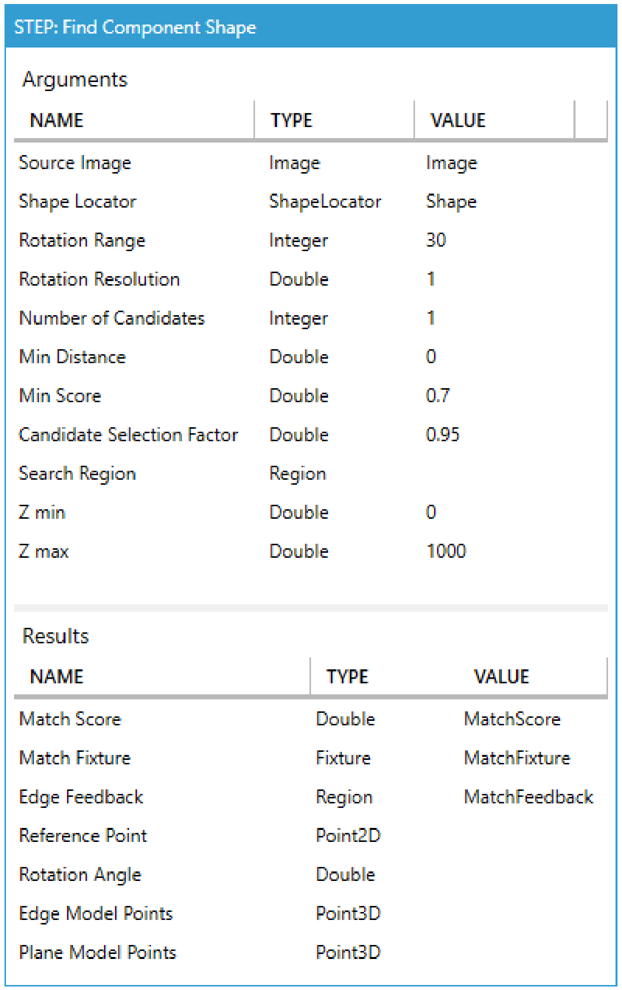
TUTORIAL 1
FIND THE COMPONENT USING MATCHING

SICK Sensor Intelligence.

- The **Find Shape** (Matching) tool is used to match the previously taught component pattern in *Image*
  - The shape was taught in a teach program which is separate from the inspection to increase performance.
- The results are: score, a **fixture** (local coordinate system) and graphical feedback (as type region)
  - The graphical feedback corresponds to the edge information from the teach image, transformed according to the match results



Load new image	Grab From File
Find Component Shape	Shape Locator Match
Transform Mounting Points	Switch Fixture



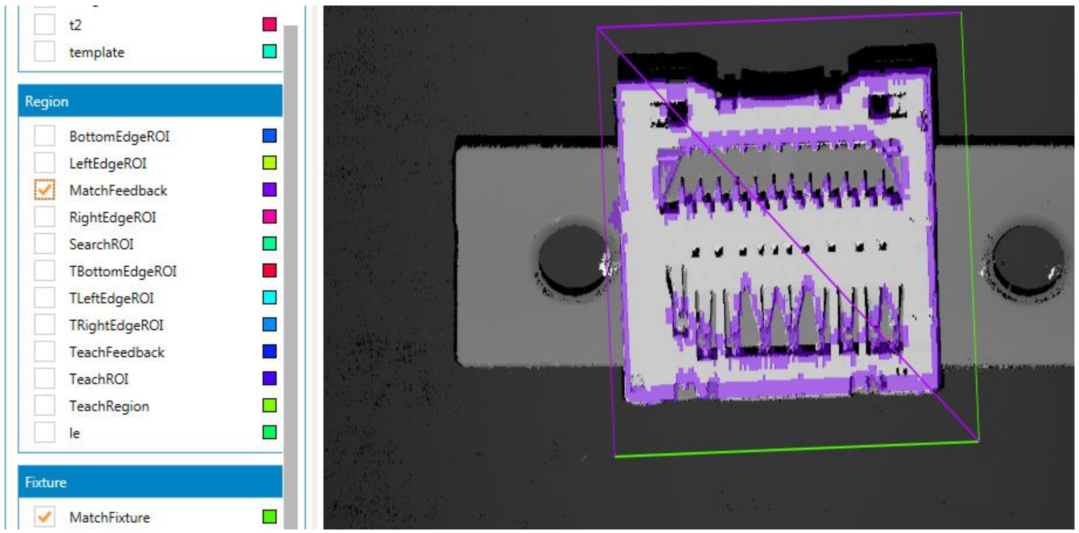
STEP: Find Component Shape

Arguments

NAME	TYPE	VALUE
Source Image	Image	Image
Shape Locator	ShapeLocator	Shape
Rotation Range	Integer	30
Rotation Resolution	Double	1
Number of Candidates	Integer	1
Min Distance	Double	0
Min Score	Double	0.7
Candidate Selection Factor	Double	0.95
Search Region	Region	
Z min	Double	0
Z max	Double	1000

Results

NAME	TYPE	VALUE
Match Score	Double	MatchScore
Match Fixture	Fixture	MatchFixture
Edge Feedback	Region	MatchFeedback
Reference Point	Point2D	
Rotation Angle	Double	
Edge Model Points	Point3D	
Plane Model Points	Point3D	



☐	t2	🟥
☐	template	🟦

Region

☐	BottomEdgeROI	🟦
☐	LeftEdgeROI	🟩
✓	MatchFeedback	🟪
☐	RightEdgeROI	🟥
☐	SearchROI	🟩
☐	TBottomEdgeROI	🟥
☐	TLeftEdgeROI	🟦
☐	TRightEdgeROI	🟦
☐	TeachFeedback	🟦
☐	TeachROI	🟪
☐	TeachRegion	🟩
☐	le	🟩

Fixture

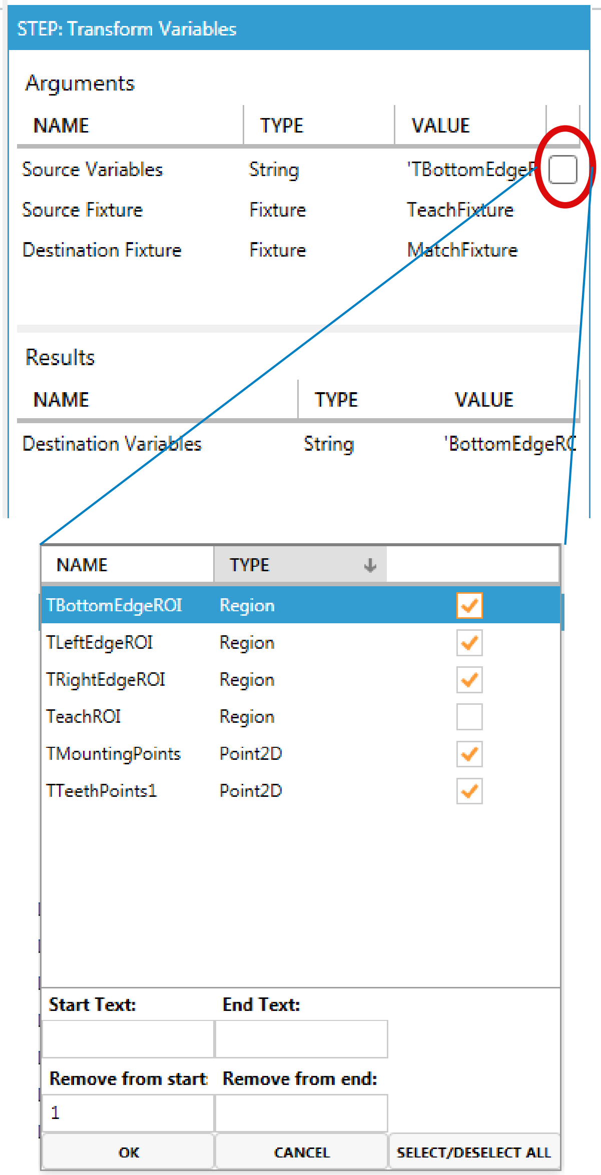
✓	MatchFixture	🟩

TUTORIAL 1
TRANSFORM POINTS AND REGIONS

- Five variables are defined (which will be transformed to the found object):

TMountingPoints	The points where the component is soldered to the PCB
TTeeth1	A set of connector pins
TBottomROI	The region around the bottom edge (for edge finding)
TLeftROI and **TRightROI**	The regions around the left and right edge

- The tool **ChangeFixture** (Coordinate Systems) is used to transform a set of points, lines or regions from one fixture to another
  - Click on the button to select which variable to transform
  - Select the names of the destination variables, in this case we just removes the T from the source variables
  - We get the originating fixture from the **Teach Shape** (Matching) tool when teaching the object.
  - We get the target fixture from **Find Shape** (Matching) when locating it in the image


STEP: Transform Variables

Arguments

NAME	TYPE	VALUE
Source Variables	String	'TBottomEdgeR
Source Fixture	Fixture	TeachFixture
Destination Fixture	Fixture	MatchFixture

Results

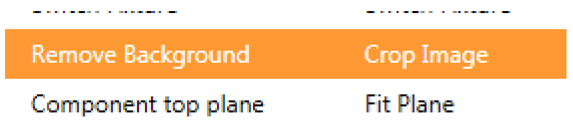
NAME	TYPE	VALUE
Destination Variables	String	'BottomEdgeRC

NAME	TYPE	
TBottomEdgeROI	Region	✓
TLeftEdgeROI	Region	✓
TRightEdgeROI	Region	✓
TeachROI	Region	
TMountingPoints	Point2D	✓
TTeethPoints1	Point2D	✓

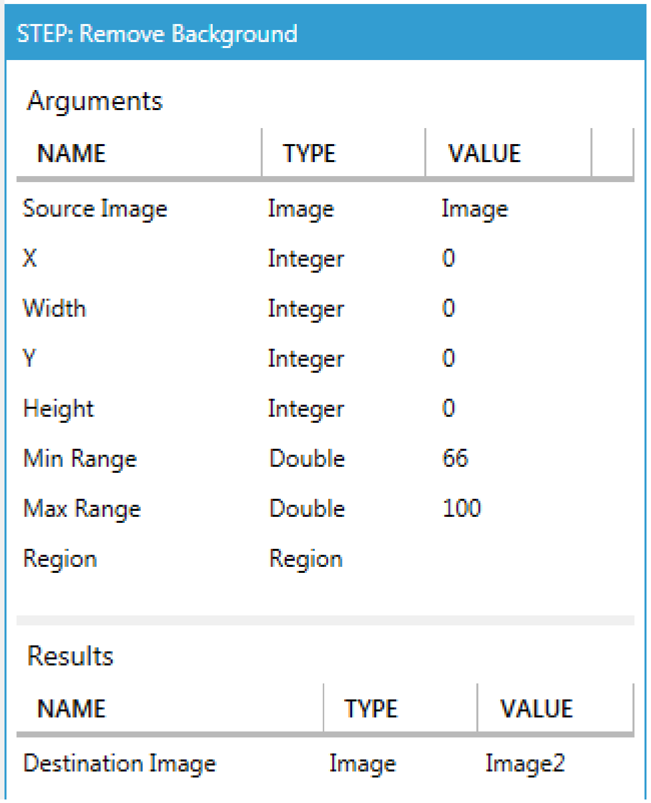
Start Text:
End Text:
Remove from start: 1
Remove from end:

TUTORIAL 1
REMOVE BACKGROUND

- Mostly for display purposes, the background data is removed from the image using the **Crop Image** (Image Operation)
- This function simply replace all data outside the Min – Max Range by *Not a Number* which is the same as *Missing Data*



Remove Background	Crop Image
Component top plane	Fit Plane



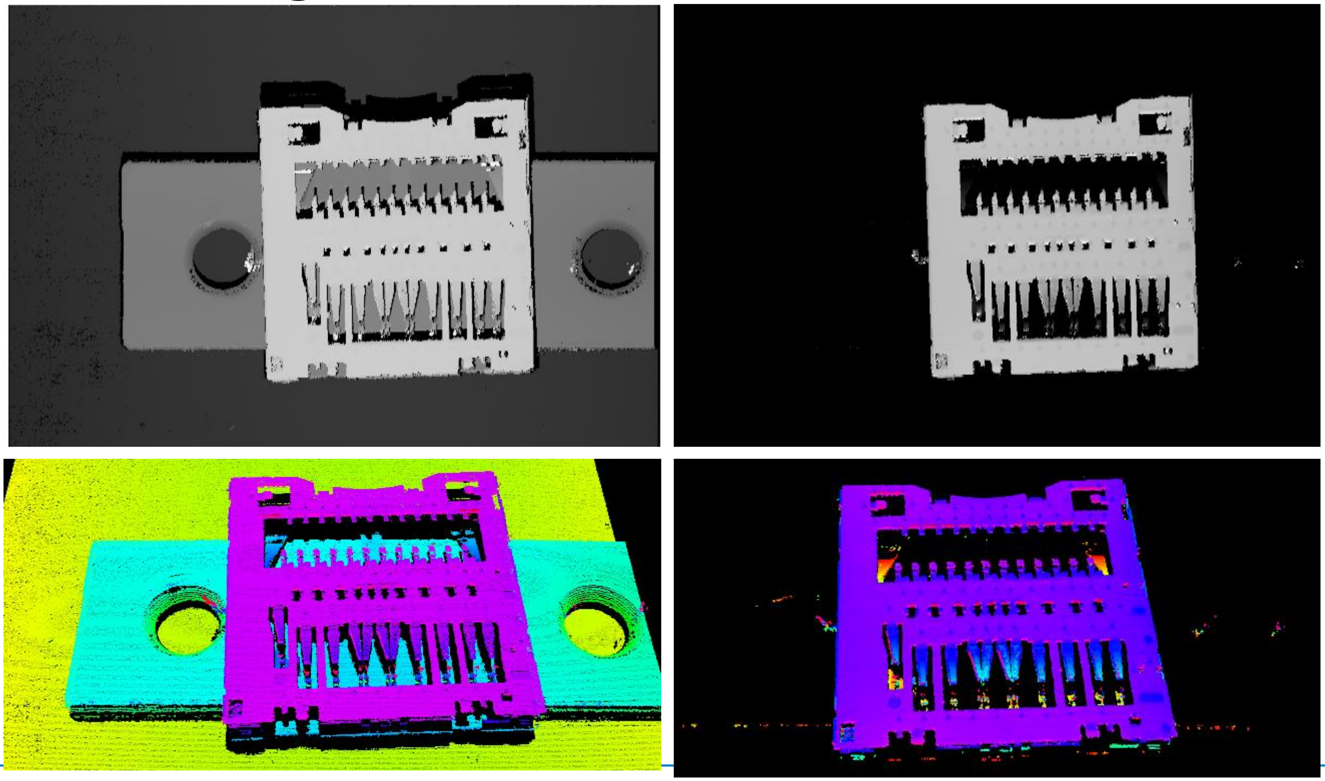
STEP: Remove Background

Arguments

NAME	TYPE	VALUE
Source Image	Image	Image
X	Integer	0
Width	Integer	0
Y	Integer	0
Height	Integer	0
Min Range	Double	66
Max Range	Double	100
Region	Region	

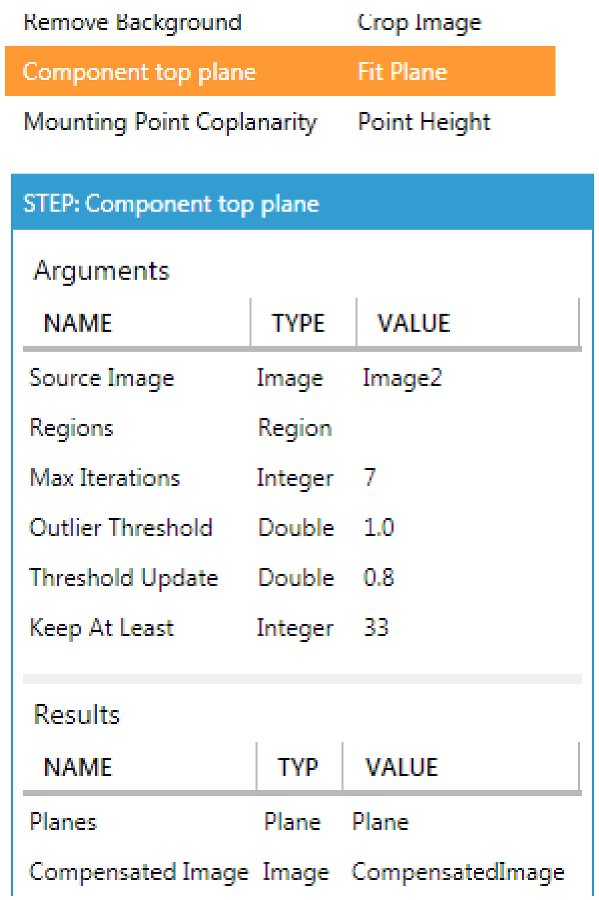
Results

NAME	TYPE	VALUE
Destination Image	Image	Image2



TUTORIAL 1
FIND COMPONENT TOP SURFACE

- The top surface of the component will be used as reference plane for the co-planarity measurements. To find this plane we use the **Fit Plane** (Feature Analysis) tool
  - A plane model is mathematically fitted to the range data in the selected region. The method is quite insensitive to noise so it is ok if there is some data part of the region but which is not in the plane. Most such noisy pixels will be iteratively filtered out.
  - The plane model can then be used in two ways:
    - As reference plane for e.g. height and volume measurements
    - To compensate the image for the plane tilt and calculate a new image by subtracting the plane level from the image
    - The CompensatedImage is an image where the top surface is flat at range value 0
  - At this stage, viewing the plane in the 3D Viewer, is a good idea to verify that the plane has been estimated correctly
    - Double click the plane and make sure that Image2 is the selected image and switch to the 3D view.



Remove Background	Crop Image
Component top plane	Fit Plane
Mounting Point Coplanarity	Point Height

STEP: Component top plane

Arguments

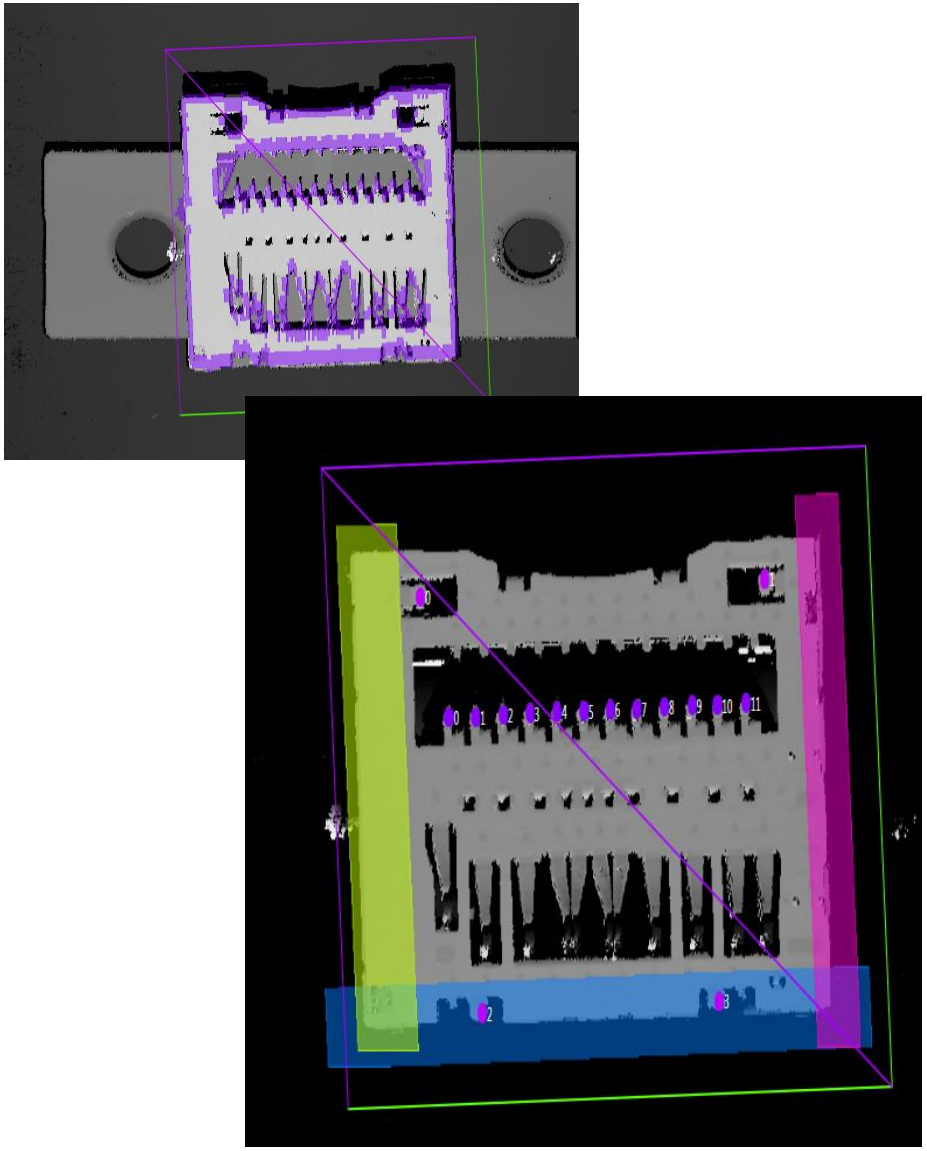
NAME	TYPE	VALUE
Source Image	Image	Image2
Regions	Region	
Max Iterations	Integer	7
Outlier Threshold	Double	1.0
Threshold Update	Double	0.8
Keep At Least	Integer	33

Results

NAME	TYP	VALUE
Planes	Plane	Plane
Compensated Image	Image	CompensatedImage

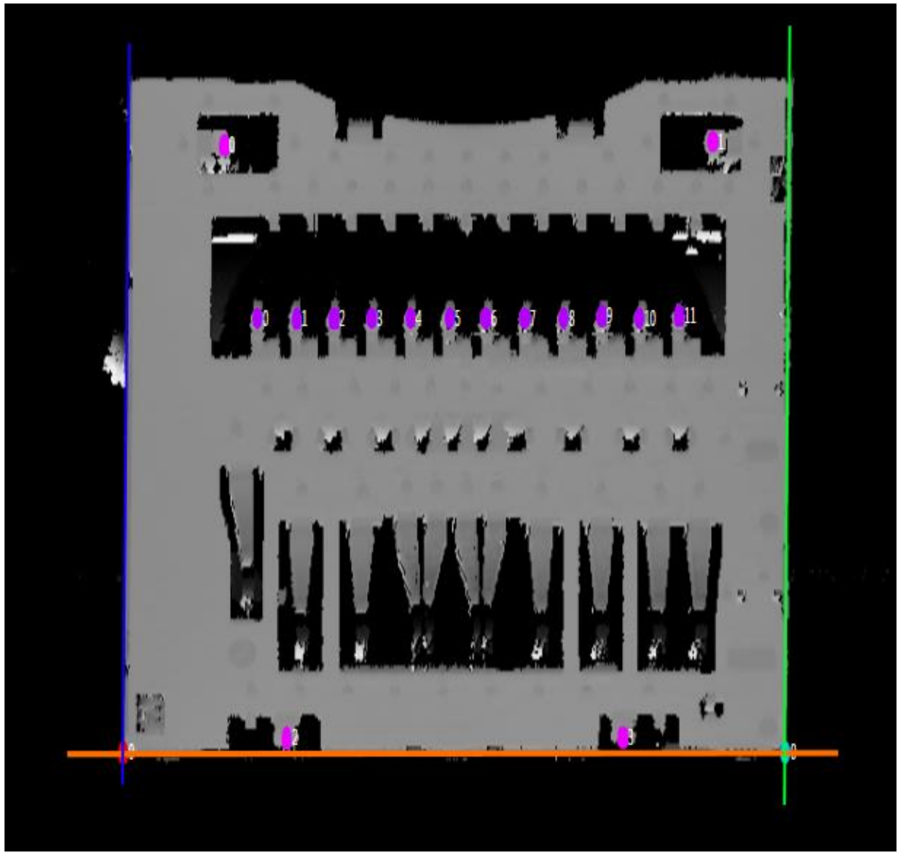
TUTORIAL 1
SUMMARY SO FAR – WHAT HAVE WE GOT NOW?

- The component is located in the image
  - The **MatchFixture** variable tells us where
- A reference plane is found in the top surface of the component
  - The CompensatedImage variable is a tilted version of the original image where this surface is flat
  - The Plane variable gives us the reference plane if we wish to use the original image data when measuring
- The measurement points and regions are transformed so they appear in the right position in the new image
  - The MountingPoints variable is a set of points in the new image corresponding to the TMountingPoints set in the template image



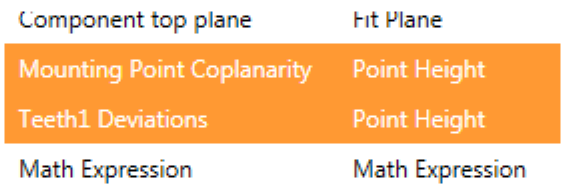
TUTORIAL 1
NEXT UP: MEASURE AND ANALYZE

- We will now measure the height of the mounting plates and connector pins relative to the component top surface
- We will also measure the width of the whole component by finding lines along the edges of the component

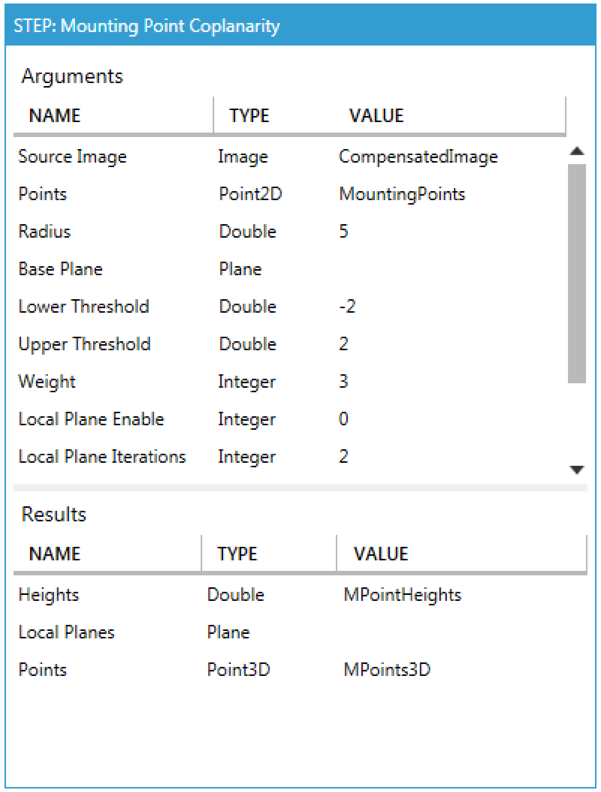


TUTORIAL 1
MEASURING PIN HEIGHTS

- To measure the height of e.g. connector pins, we use the **Point Height** (Feature Analysis) tool
  - Essentially, the tool takes a set of Point2D and an image. For each point, it evaluates the height in the image.
- The tool allows two kinds of smart probing;
  - Local averaging. The average height value in the region around the point is calculated. You can select the size of the region and how to give different weights to values depending on their distance to the point
  - Local plane fitting. A mathematical plane is fitted to the region around each point
  - See the manual for details on the tool's many parameters
- The heights returned are the vertical distances to the reference plane (Base Plane).
  - No base plane means the distance from the zero plane.
  - In our case we use the zero plane and the plane compensated image.
- The tool can return a set of height values (one per point), but also a Point3D set which can e.g. be used as input to a **Plane From Points** (Feature Analysis) tool.



Component top plane	Fit Plane
Mounting Point Coplanarity	Point Height
Teeth1 Deviations	Point Height
Math Expression	Math Expression



STEP: Mounting Point Coplanarity

Arguments

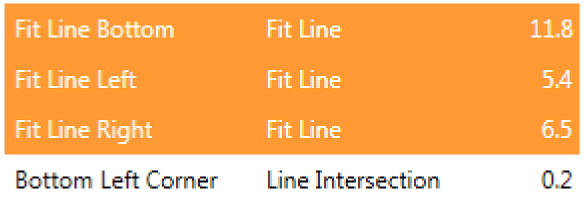
NAME	TYPE	VALUE
Source Image	Image	CompensatedImage
Points	Point2D	MountingPoints
Radius	Double	5
Base Plane	Plane	
Lower Threshold	Double	-2
Upper Threshold	Double	2
Weight	Integer	3
Local Plane Enable	Integer	0
Local Plane Iterations	Integer	2

Results

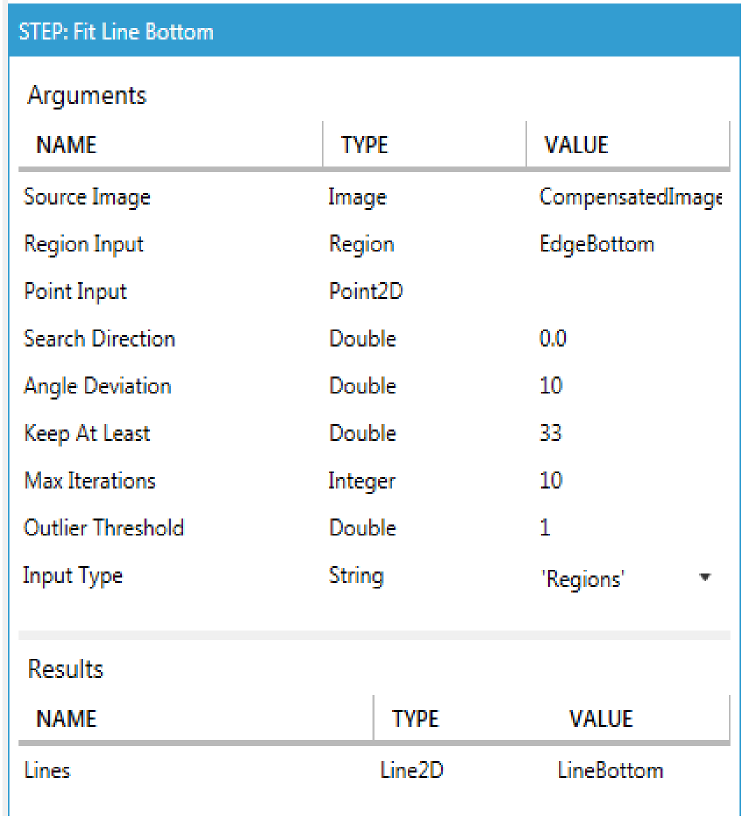
NAME	TYPE	VALUE
Heights	Double	MPointHeights
Local Planes	Plane	
Points	Point3D	MPoints3D

TUTORIAL 1
FIND COMPONENT LINES

- Using **Fit Line** (Feature Analysis), a line is fitted to the edges insided the input region.
- One region per line is used
  - The bottom edge is located using a region around the bottom edge of the object. That region we have previously drawn in the template image and transformed to the position of the current image. We locate the left and right edges using the same technique.



Fit Line Bottom	Fit Line	11.8
Fit Line Left	Fit Line	5.4
Fit Line Right	Fit Line	6.5
Bottom Left Corner	Line Intersection	0.2



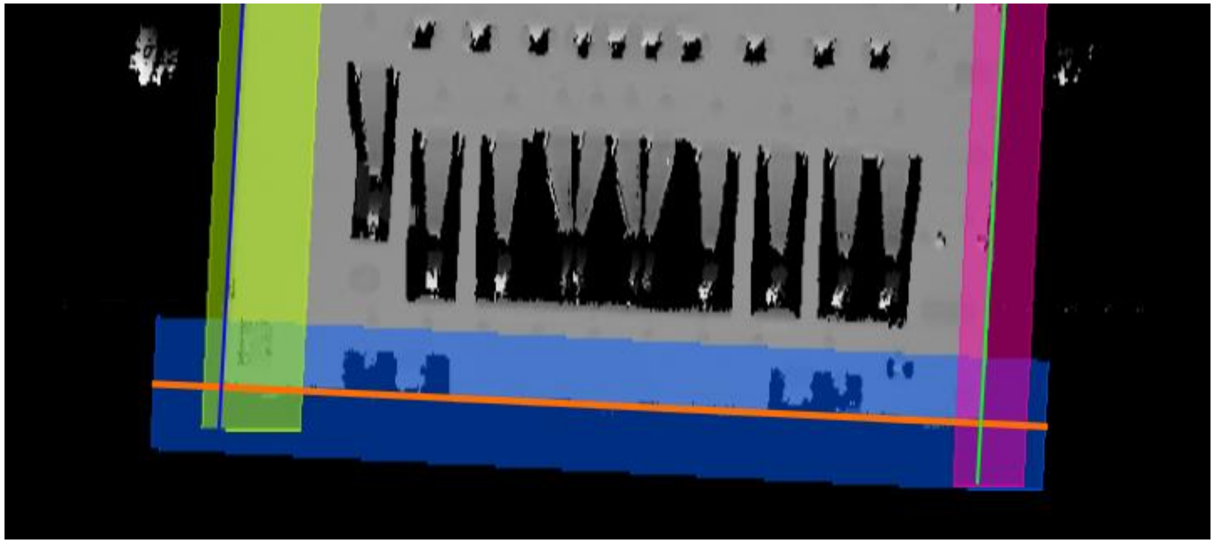
STEP: Fit Line Bottom

Arguments

NAME	TYPE	VALUE
Source Image	Image	CompensatedImage
Region Input	Region	EdgeBottom
Point Input	Point2D	
Search Direction	Double	0.0
Angle Deviation	Double	10
Keep At Least	Double	33
Max Iterations	Integer	10
Outlier Threshold	Double	1
Input Type	String	'Regions'

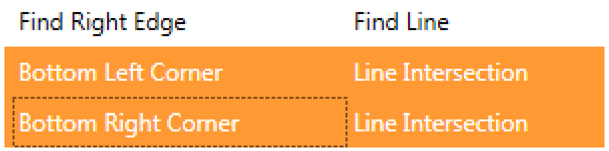
Results

NAME	TYPE	VALUE
Lines	Line2D	LineBottom

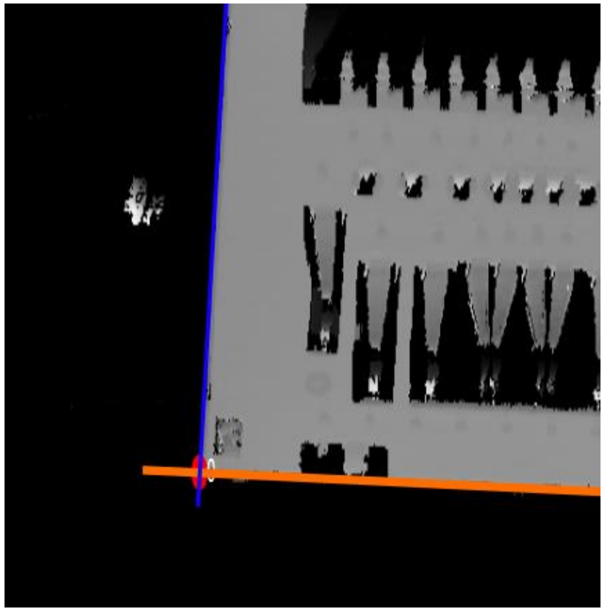


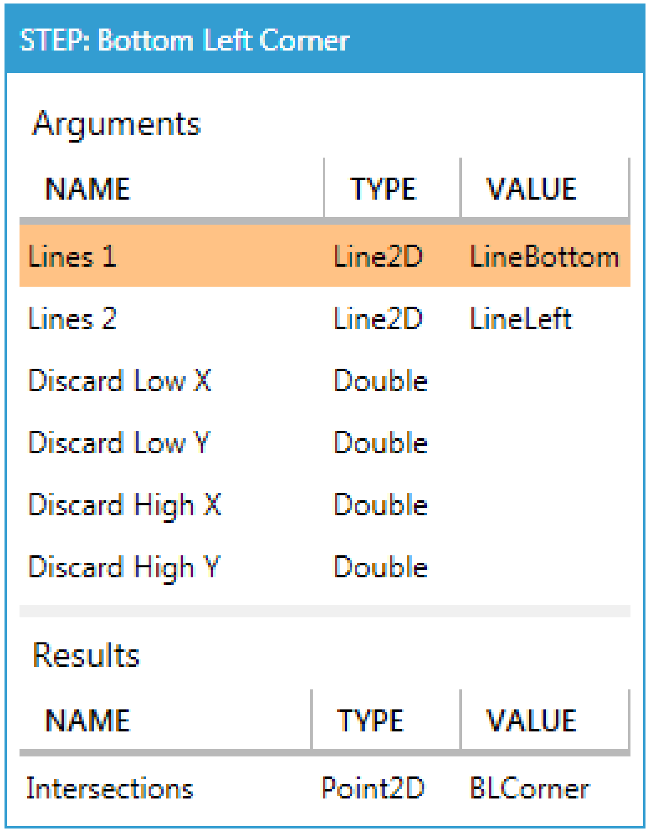
TUTORIAL 1
FIND CORNERS AS INTERSECTION POINTS

- To measure the width of the object, the corners of the object are located.
  - The corners are defined as the intersection points between adjacent edge lines
  - The **LineIntersection** (Feature Analysis) tool calculates the position where two lines cross
  - The result is a Point2D at the intersection



Find Right Edge	Find Line
Bottom Left Corner	Line Intersection
Bottom Right Corner	Line Intersection





STEP: Bottom Left Corner

Arguments

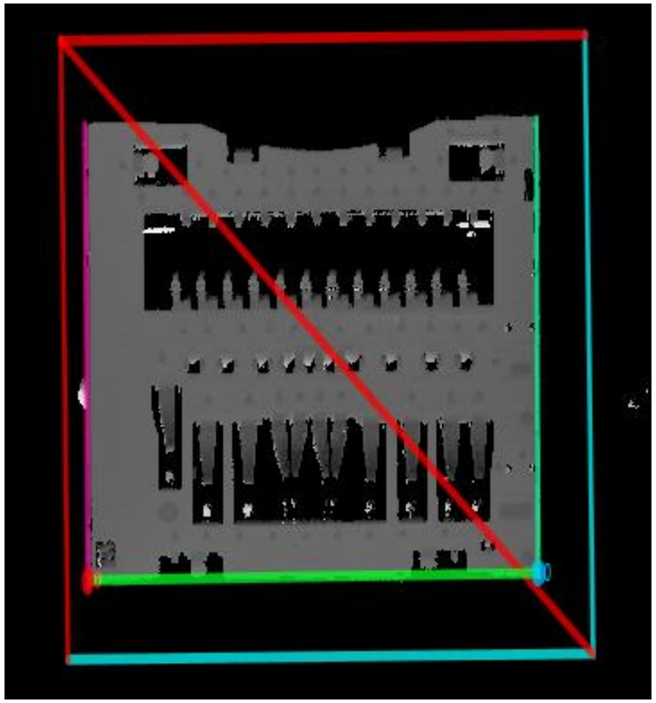
NAME	TYPE	VALUE
Lines 1	Line2D	LineBottom
Lines 2	Line2D	LineLeft
Discard Low X	Double	
Discard Low Y	Double	
Discard High X	Double	
Discard High Y	Double	

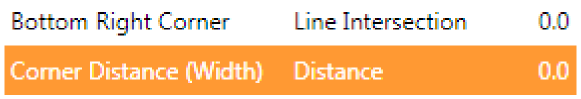
Results

NAME	TYPE	VALUE
Intersections	Point2D	BLCorner

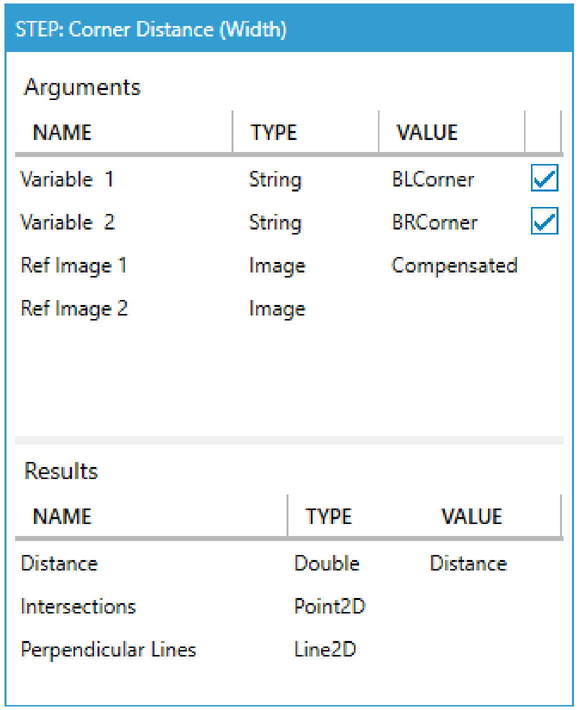
TUTORIAL 1
WIDTH AS CORNER DISTANCE

- Finally, the width of the component as the distance between the bottom left and bottom right corner is calculated
  - The **Distance** tool (Feature Analysis) together with the two corner points we calculated are used
- Click **Run Continuously** to loop through the set of images





Bottom Right Corner	Line Intersection	0.0
Corner Distance (Width)	Distance	0.0



STEP: Corner Distance (Width)

Arguments

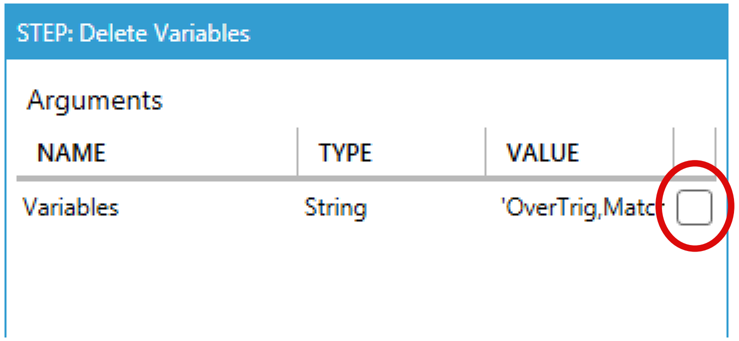
NAME	TYPE	VALUE	
Variable 1	String	BLCorner	✓
Variable 2	String	BRCorner	✓
Ref Image 1	Image	Compensated	
Ref Image 2	Image		

Results

NAME	TYPE	VALUE
Distance	Double	Distance
Intersections	Point2D	
Perpendicular Lines	Line2D	

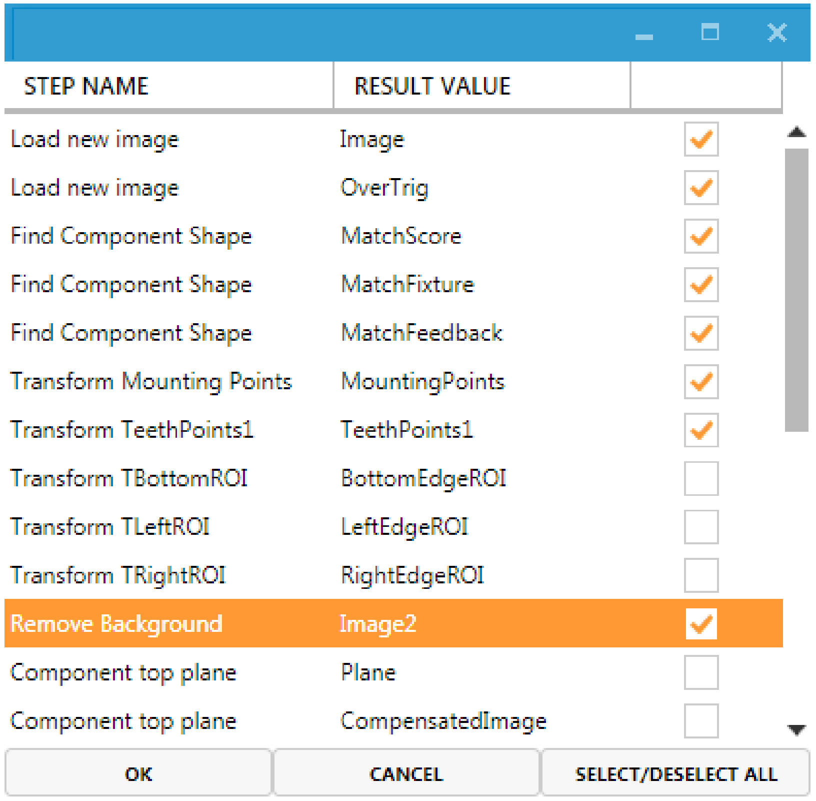
TUTORIAL 1
DELETE VARIABLES

- At the end of each program it can be useful to add a Delete Variables step to remove intermediate variables. This is especially helpful when using multiple programs and you want to keep clean structure of the environment and get an better overview of the variable list
  - Right-click in the step program view and select **Create Delete Step**
  - Click on the button next to the input argument, you will get a list of the filled out output variables from all the steps in the current step program
  - Select the variables you want to delete and click **OK**
  - When debugging systems, you might want to keep all variables. You can disable an individual step by right-clicking and press **Enable/Disable**


STEP: Delete Variables

Arguments

NAME	TYPE	VALUE	
Variables	String	'OverTrig,Matc	☐



STEP NAME	RESULT VALUE	
Load new image	Image	✓
Load new image	OverTrig	✓
Find Component Shape	MatchScore	✓
Find Component Shape	MatchFixture	✓
Find Component Shape	MatchFeedback	✓
Transform Mounting Points	MountingPoints	✓
Transform TeethPoints1	TeethPoints1	✓
Transform TBottomROI	BottomEdgeROI	☐
Transform TLeftROI	LeftEdgeROI	☐
Transform TRightROI	RightEdgeROI	☐
Remove Background	Image2	✓
Component top plane	Plane	☐
Component top plane	CompensatedImage	☐

EASY RANGER
MORE INFORMATION

- Go to Support Portal for more information and tutorials!
  - https://supportportal.sick.com/downloads/easy-ranger

Sensor Intelligence.