NEURO CHECK
Industrial Vision Systems

Application Software for Industrial Vision Systems

Roi Tools

Plug-In

Version 6.2.10
Neckarstraße 76-1, 71686 Remseck, Germany

Phone: +49 (0) 7146 - 89 56-0
Fax: +49 (0) 7146 - 89 56-29
E-Mail: info@neurocheck.com
Web: www.neurocheck.com

Table of Contents

NeuroCheck PI_RoiTools.NET Plug-In Help 4
General Information 4
Introduction 4
Installation 5
Check Functions 6
Change Group Numbers 6
Introduction 6
Parameter Dialog 7
Visualization 8
Check ROIs in Target Area 9
Introduction 9
How to Use 11
Parameter Dialog 12
Visualization 15
Concatenate ROIs 16
Introduction 16
Parameter Dialog 19
Create Bounding Box 21
Introduction 21
How to Use 24
Parameter Dialog 33
Create Positioned ROIs 34
Introduction 34
How to Use 35
Parameter Dialog 36
Visualization 38
Create ROIs from Model Geometries 39
Introduction 39
How to Use 40
Parameter Dialog 41
Visualization 43
Load ROIs 44
Introduction 44
Parameter Dialog 45
Load ROIs from Directory 46
Introduction 46
Parameter Dialog 47
Merge ROIs 49
Introduction 49
How to Use 50
Parameter Dialog 52
Visualization 53
Modify ROIs Contour 54

Introduction 54
How to Use 55
Parameter Dialog 57
Visualization 58
Reverse Transform Unrolled ROIs 59
Introduction 59
How to Use 61
Parameter Dialog 64
Visualization 65
Save ROIs 66
Introduction 66
Parameter Dialog 67
Tile ROIs 68
Introduction 68
Parameter Dialog 69
Visualization 70
Support Contact 71
Support Services 71

Introduction

General information about NeuroCheck plug-in DLLs

A plug-in DLL is a .NET assembly that serves to enhance NeuroCheck with user-defined image processing functionality. The NeuroCheck Plug-In Interface offers the opportunity to integrate user-defined check functions for image processing and data handling. A Plug-In can contain an arbitrary number of self-developed check functions.

These check functions have full access to the NeuroCheck runtime data objects such as Images, ROI Lists or Measurement Lists. The Plug-In check function can be added to a check as well as the built-in standard check functions of NeuroCheck.

Please note that for integration of a plug-in check function into your check routine, a Premium license is required. The completed check routine then can be run with any NeuroCheck license (except the Demo Version).

Installation

Installation

Copy the following files from the zip archive into the Plug-In directory within the desired NeuroCheck project, e.g. `'C:\Users\Public\Documents\NeuroCheck\6.2\Default\Software Extensions\PlugIns\'`.

- `XX\PI_RoiTools.NET.resources.dll` (XX stands for the 'ISO 2 Letter Language Code', e.g. de)
- `PI_RoiTools.NET.chm`
- `PI_RoiTools.NET.dll`

Loading a Plug-In

In order to use a Plug-In the Plug-In assembly must be loaded in NeuroCheck. The management of Plug-Ins takes place within the Software Settings dialog. The Software Settings dialog can be found in the System menu of NeuroCheck.

Please note that it is impossible to load or unload a Plug-In as long as a check routine is opened that contains the Plug-In check functions. If the currently opened check routine contains Plug-In check functions then close the check routine first.

Within the Software Settings dialog please select the node Plug-Ins and the sub-node Plug-In in the tree to the left. The loaded Plug-In assemblies are shown in the List of Plug-Ins. Press the Add button to open a file selection dialog in order to select a further Plug-In assembly.

Inserting a Plug-In check function to a check routine

A Plug-In check function is inserted using the Check Function Select dialog. All check functions of loaded Plug-Ins are listed in the Plug-In category of the Check Function Select dialog. Within the Plug-In category the check functions are ordered in sub-categories where each sub-category represents the check functions of one Plug-In.

Besides the category the user will hardly notice any difference between the usage of Plug-In check functions and built-in check functions.

Change Group Numbers: Introduction

Function overview

This function can be used to change the group numbers for the objects in the input list of ROIs. Possible change modes are fix group numbers, group numbers per class and different group number for each AOI.

Input data

This check function requires an image and a list of ROIs as input data objects.

Output data

None (changed group numbers in input list of ROIs).

Result view

The result view shows the input image and a list of the input ROI collection.

Properties

 Check function group Plug-In.

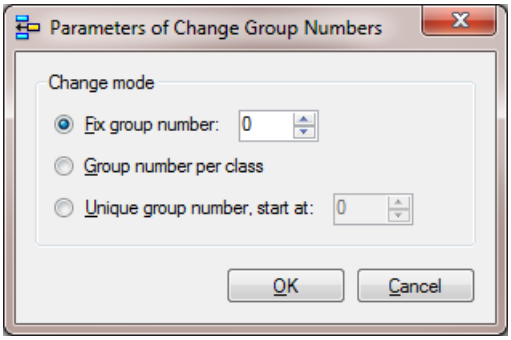
 The check function has a Parameter Dialog.

 The check function has own result Visualizations.

Change Group Numbers: Parameter Dialog

This plug-in check function has a **Parameter** dialog.

Screenshot of Parameter Dialog



The **Parameter** dialog contains the following elements:

Element	Description
Fix group numbers	All ROIs get the same number as specified in the parameter dialog.
Group number per class	All ROIs belonging to the same class will get the same group number (=class ID).
Unique group number, start at	All ROIs will get a different number starting with the number as specified in the parameter dialog.

Change Group Numbers: Visualization

This section describes the result visualizations the check function "Change Group Numbers" provides.

Element	Description
Input Image	Displays the input image of the check function.
Input ROI Collection	Displays a list of the input ROI collection data of the check function.

Check ROIs in Target Area: Introduction

Function overview

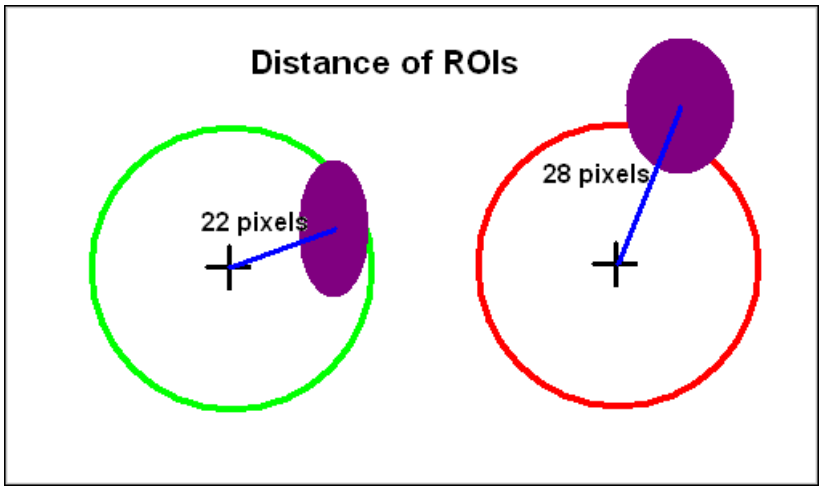
This check function can be used to verify if for each search ROI exactly one object (part) exist.
It's also possible to display regions where parts are missing or have been displaced.

The first list of ROIs contains the "search ROIs" (for instance manually defined ROIs), the second list of ROIs the "target ROIs" (for instance binary created ROIs). For each search object, the nearest target object (in the other list of ROIs) is searched which belongs to the same group. The matching ROIs from both lists form a pair. A quality is calculated for each pair.

There are two methods for calculating this quality:

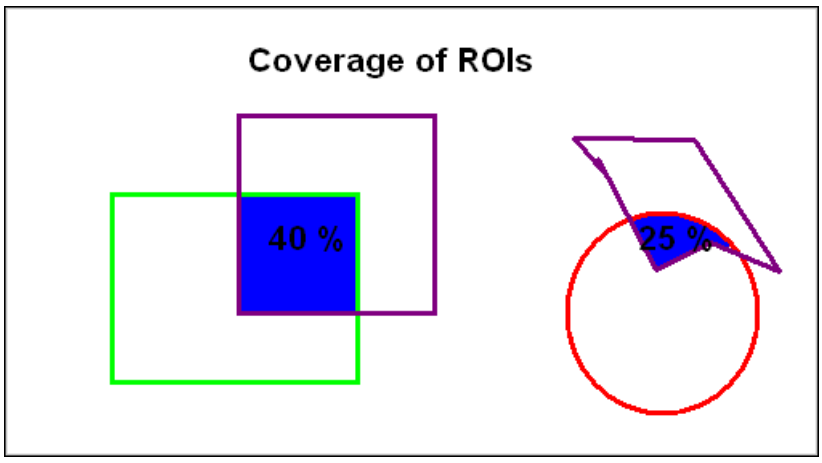
- the distance of center points in pixels.

  See an example



- the coverage of regions as percentage (e.g. measurement value 0.4 means 40%).

  See an example



The coverage is defined as ratio of area of intersection between search and target ROI with respect to the area of the target ROI. If the target ROI is completely contained in the search ROI, the ratio will be 1, i.e. 100%.In this mode AOIs and binary created objects can be used.

The quality (distance or coverage value) is assigned as new feature value for both search and target ROIs. Furthermore, for each search ROI a new measurement gives the quality (distance or coverage ratio) in the output measurement list. The number of measurements in the output list corresponds to the number of search ROIs (usually fixed number). This can be used for further processing e.g. with Check Allowance. A measurement will be assigned even if no target object was found for a specific search object, so that the number of output measurements is constant.

Input data

This check function requires an image and two collections of ROIs as input data objects.

- Image
- List of ROIs (Search regions)
- List of ROIs (Target objects)

Result view

The result view shows the search and target objects. In the second tab it shows only the search objects. It is possible to set options for the result view in the parameter dialog.

Properties

 Check function group Plug-In.

 The check function has a Parameter Dialog.

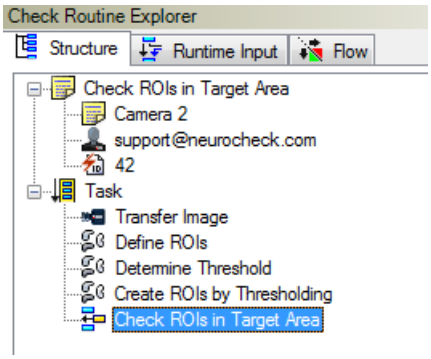
 The check function has own result Visualizations.

Check ROIs in Target Area: How to Use

The check function compares two lists of ROIs by building pairs of ROIs.
Normally a fix positioned list of ROIs is used as search ROIs. The target ROIs are binary created, whose existence and position depends on the quality of the good or bad part.

Check routine sample

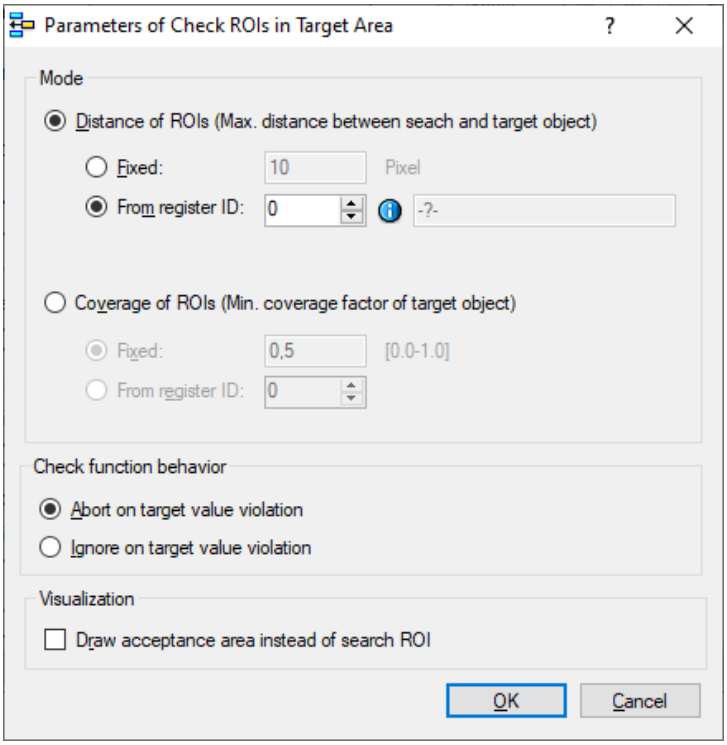
Screenshot of Check Routine Example



Check ROIs in Target Area: Parameter Dialog

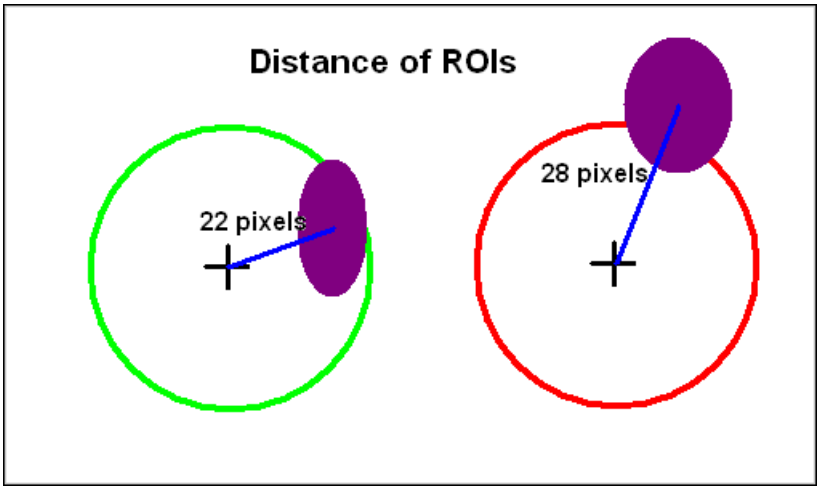
This plug-in check function has a **Parameter** dialog.

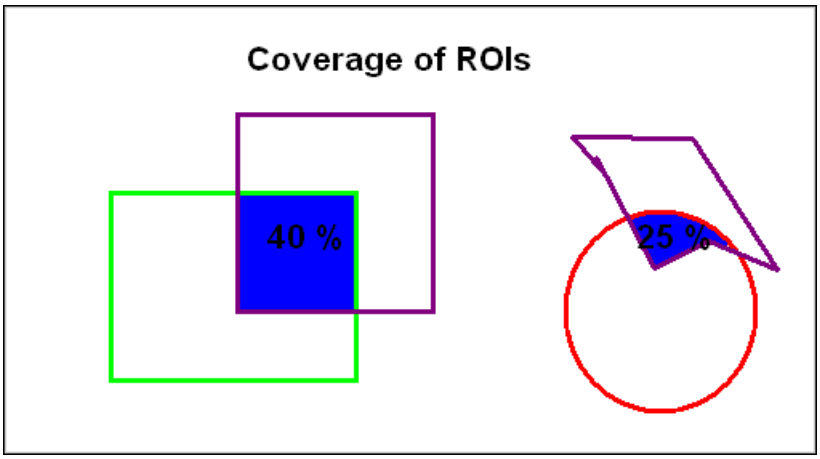
Screenshot of Parameter Dialog



The **Parameter** dialog contains the following elements:

Element	Description
Mode	There are two ways of analyzing methods. A change here implies the adjustment of the input measurement because it will be interpreted in a different way.

Element	Description
Max. distance between search and target object	A circle is defined around the center point of the search object with the radius (distance) defined by the input measurement. If no different groups are selected, one measurement in the input measurement list will be enough. If the search objects are created in groups, every group of search objects will have its own limit of distance as measurement (radius in pixels).If the distance between the center points of search object and target object of a pair is less than the specified limit (radius), the object is present and the check is OK. See an example  • **Fixed**: The check function will always use the specified value. • **From register ID**: The check function determines the value (dynamically) from the Data Register Manager using the specified register ID. The register cell must be of type integer or floating point number.

Element	Description
Min. coverage factor of target object	The search object is defined as region of the input ROI, the input ROIs draws the search object, a target object is expected in. The coverage of the pair is calculated as percentage of the intersection of the ROI regions compared to the region of the target object. If this coverage is larger than the limit in the measurement list (given as percentage, e.g. 0.4 means 40%), the check is OK. If there are multiple groups in the input list, every group has its own limit in the measurement list. See an example  • **Fixed**: The check function will always use the specified value. • **From register ID**: The check function determines the value (dynamically) from the Data Register Manager using the specified register ID. The register cell must be of type floating point number.
Abort on target value violation	If one of the objects is out of range or not present, the whole check function fails.
Ignore on target value violation	The check function only displays the errors using the colored contours in the result view.
Visualization	The check function provides a number of result views. The visualization is quite important for this check function because it can be used to indicate "missing" which is not possible with NeuroCheck standard check function. Here you can configure some aspects of the presentation in the result views.
Display index of ROIs	If activated, then next to the search region displayed in the result views "Search objects" and "Search and target objects" the index of each search region will be displayed. This allows an easier association of the results provided in the measurement list to the search objects. However, in some cases the displayed index is rather bothering because it covers important parts of the image. Therefore it is possible to deactivate the display of the index.
Draw acceptance area instead of search ROI	This setting only has an effect if the "Distance" mode is chosen for Quality. In this case the acceptance area is the circle with radius taken from the input measurement and center point identical to that of the search ROI. The center point of the target object must be inside this circle, therefore it is in many cases more convenient to display this circle instead of the search object. However, if the radius for the circle is quite small, it might only be visible as small dot in the image. In this case, it might be of advantage to display the search object. You can choose which visualization suits more your purpose.

Check ROIs in Target Area: Visualizations

This section describes the result visualizations the check function "Check ROIs in Target Area" provides.

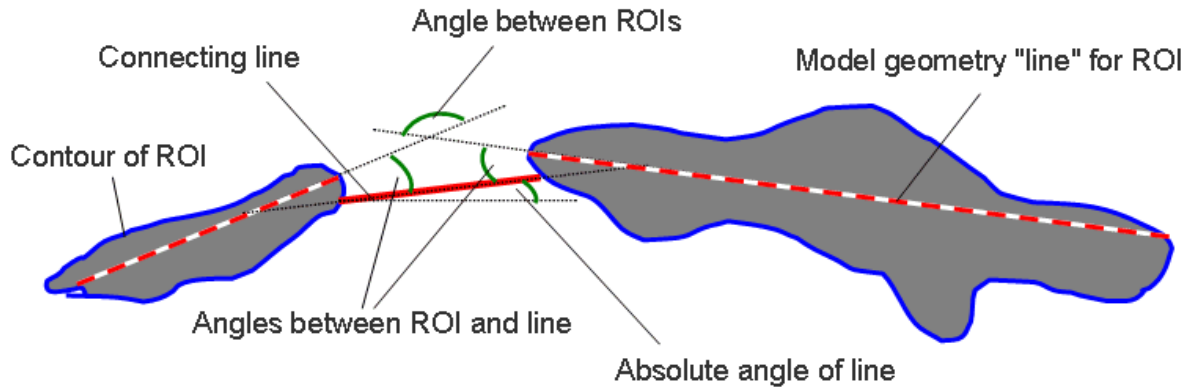
Element	Description
Search and Target objects	This visualization shows both search and target ROIs in one view.
Search objects	This visualization shows only search ROIs in one view.

Concatenate ROIs: Introduction

Function

This plug-In check function evaluates the contours of the input ROIs and creates a new list of ROIs with lines where the contours of the input ROIs are connected. It also creates an image with these new ROIs drawn into it. To get the connected ROIs, use the standard check function "Create ROIs by Thresholding".
In addition to creating the connecting lines, the function computes a number of feature values which are attached to the original ROIs and to the resulting connecting lines. For the computed feature values, please see section "Computed Feature Values" below.

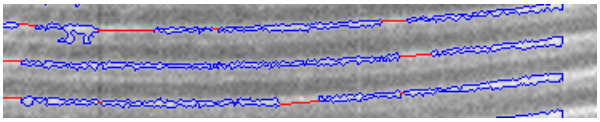
The following is an illustration to show the computed angles:



Possible applications

Concatenation

This function can be used to concatenate ROIs if an object breaks into smaller pieces during binary segmentation. For instance, it could be used in analyzing scratches during a surface inspection. If the scratches break into smaller pieces, they might be accepted as small errors. This function can be used to find out if the smaller parts all belong to a larger object.



Measurement of distance

This function computes the minimum distance between the contours of ROIs equal to "Gauge ROIs" with rule "Distance, minimal contour-contour". Thus it can be applied if the minimum distance of a varying number of objects must be evaluated. The distances are attached as feature values both to the original ROIs and to the connecting lines.

Requirements

- The check function needs ROIs that have a contour (e.g. binary objects from Create ROIs by Thresholding) or rectangular AOIs (e.g. results from Template Matching). Mixing contour objects and AOIs will not work.
- If the conditions "Angle between ROIs" or "Angle between connecting line and ROIs" are used, each input ROI needs (depending on your choice in the parameter dialog) either a model geometry line or the following features: Center X, Center Y, Axis length major, Axis orientation [0..360].
- Information: Objects which have a distance of 2 or less pixels are not considered for angle checking.

Computed Feature Values

The following feature values are computed for the resulting connecting lines:

ID	Name	Meaning	Notes

4301	Distance	Distance of the two ROIs that this line connects	
4302	Index ROI#1	Index of first ROI	
4303	Index ROI#2	Index of second ROI	
4304	Angle ROIs	Angle between lines for angle computation of the two ROIs	Range 0 - 180 degrees
4305	Angle ROI#1	Angle between connecting line and line for angle computation of the first ROI	Range 0 - 90 degrees
4306	Angle ROI#2	Angle between connecting line and line for angle computation of the second ROI	Range 0 - 90 degrees
4307	Angle Line	Absolute angle of connecting line in relation to horizontal image border	Range -90 - 90 degrees

The following feature values are computed for the input ROIs:

ID	Name	Meaning	Notes
4308	Angle nearest ROI	Angle between lines for angle computation	-1 means no connection found
4309	Distance nearest ROI	Distance to nearest ROI	-1 means no connection found
4310	Index nearest ROI	Index of nearest ROI	-1 means no connection found
4311	Angle ROI connecting line to nearest ROI	Angle between connecting line and line for angle computation	-1 means no connection found
4312	Angle second nearest ROI	Angle between lines for angle computation	-1 means no connection found
4313	Distance second nearest ROI	Distance to second nearest ROI	-1 means no connection found
4314	Index second nearest ROI	Index of second nearest ROI	-1 means no connection found
4315	Angle ROI connecting line to second nearest ROI	Angle between connecting line and line for angle computation	-1 means no connection found

Input data

This check function requires an image and a list of ROIs input data objects. The image is used to visualize the results and the list of ROIs contains the ROIs that should be concatenated.

Output data

The check function has an image and a list of ROIs as output data objects. The image shows the created line ROIs and the list of ROIs contains these lines.

Result view

The check function has three result views. The first shows the input ROIs and the created lines in a copy of the input image, while the second view only shows the lines and the third view only the input ROIs.

In case of an error, the error message is displayed in the result view.

Properties

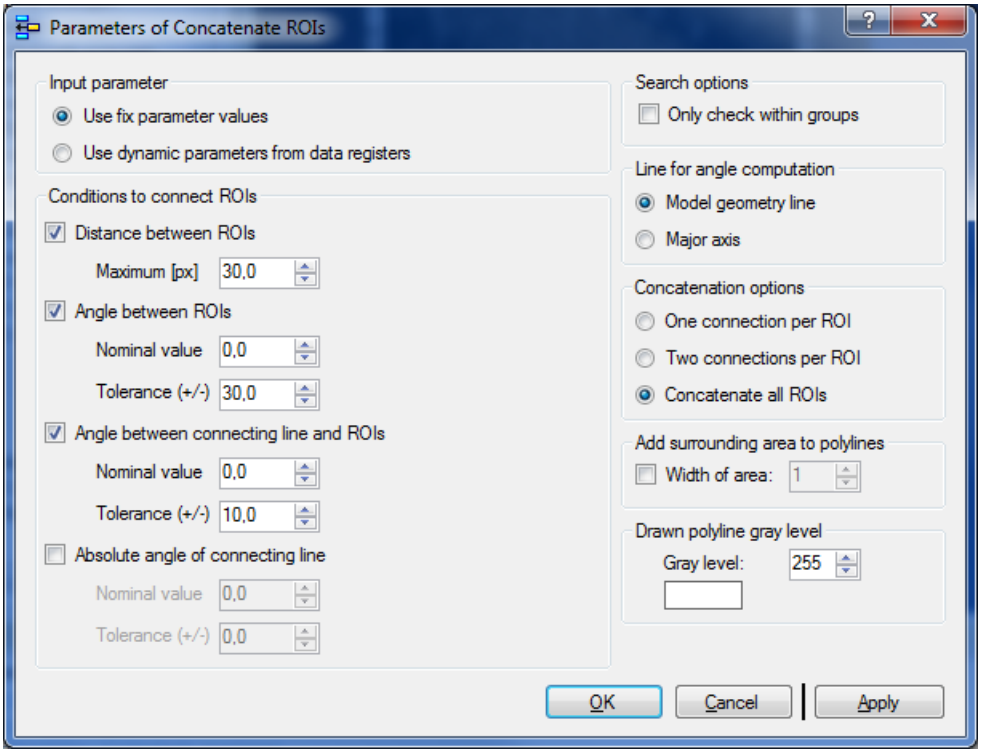
 Check function group Plug-In

 The check function has a Parameter Dialog.

Concatenate ROIs: Parameter Dialog

This check function has a **Parameter** dialog.

Screenshot of Parameter Dialog



The **Parameter** dialog contains the following elements:

Element	Description
Input Parameter	Set the data source of the input parameters.
Use fix parameter values	Select this if the values for the conditions should be fix entered in the parameter dialog.
Use dynamic parameters from data registers	Select this if you want to load the values for the conditions dynamically from register cells.
Conditions to connect ROIs	Specify the conditions that need to be met by two ROIs so they will be concatenated.
Distance between ROIs	Specify the maximum distance two ROIs are allowed to have in pixel.
Angle between ROIs	Specify the angle between ROIs that needs to be satisfied. This is useful if you wish to connect ROIs which are approximately "in line" (angle close to 180 degrees), but not perpendicular (angle of 90 degrees). Please note that there is no direction of the model geometry "lines". Thus it is difficult to distinguish for "overlapping" lines if they are "parallel" or "in line", i.e. if the angle is closer to 0 or closer to 180 degrees.

Element	Description
Angle between connecting line and ROIs	Specify the angle a connection between two ROIs (or better, the model geometry line calculated for the ROI) must satisfy the specified tolerances. Use this option, if you know that the connecting ROIs are approximately "in line", so the angle between the connection line and the ROIs should be around 0 degrees. If you know that they are slightly bent (following a curvature), this angle can vary. A problem here is that you cannot decide in which direction (left or right) the curvature is followed, there is no decision which of the two connected ROIs really is the first one.
Absolute angle of connecting line	Specify the angle (relative to the image border) that two ROIs must satisfy with the specified tolerances. Use this option, if you know the approximate direction the connections in the image should have (e.g. only horizontal connections = angle of zero degrees).
Search options	Select options which affect the search process for new connections.
Only check within groups	Select this if your ROIs are grouped and you want to search for connections only in ROIs that are in the same group.
Line for angle computation	Select the line that is used to compute the angles.
Model geometry line	Select this if you want to use the model geometry line for the angle computation.
Major axis	Select this if you want to use the major axis line for the angle computation. The following features need to be computed for each ROI before executing this check function: Center X, Center Y, Axis length major, Axis orientation [0..360].
Concatenation options	Select how many connections you want per ROI.
One connection per ROI	Select this if you want each ROI to be connected only to its nearest neighbor.
Two connections per ROI	Select this if you want each ROI to be connected to its two nearest neighbors.
Concatenate all ROIs	Select this if you want to connect all ROIs that satisfy the conditions.
Add surrounding area to polylines	If you want to add a surrounding area (as in Define ROIs) to the resulting lines, you can do it here. This is useful if you wish to create thicker lines in the resulting image with the drawn lines. Specify the width in pixel.
Drawn polyline gray level	If you want to change the gray level of the lines in the result image, you can set the value here.

Create Bounding Box: Introduction

Function overview

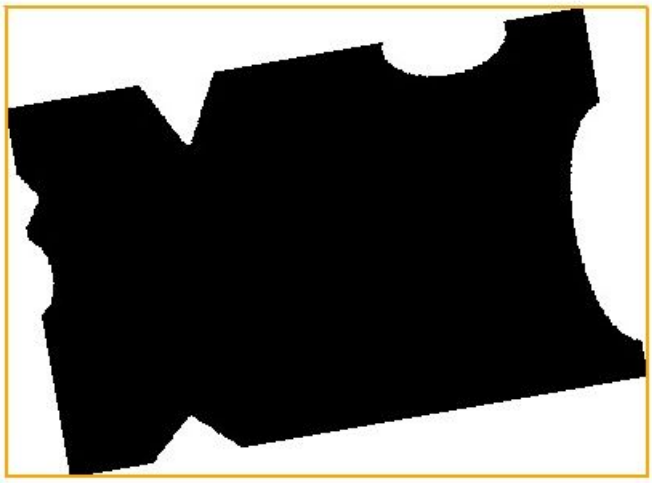
This check function calculates a Bounding Box for each object or each group in the input list of ROIs. There are two Bounding box modes available: Outer Bounding Box and Inner Bounding Box.

Definition

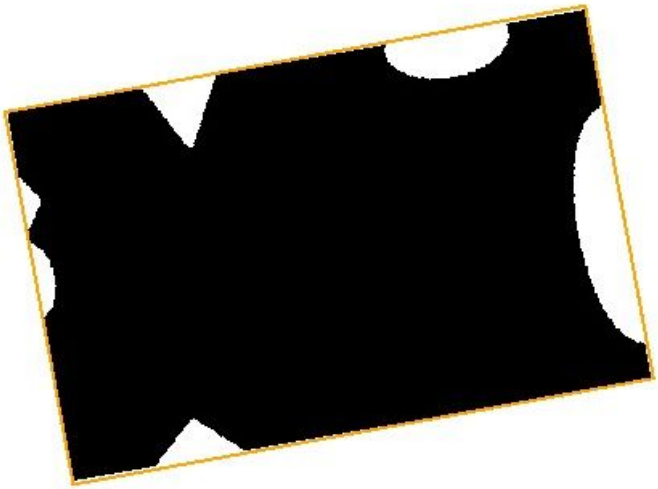
The Outer Bounding Box is defined as the smallest rectangle enclosing a given object contour. Accordingly the Inner Bounding Box in general is the largest rectangle enclosed by the object contour. Depending on the chosen parameter for angle and raster size, the Bounding Box may not necessarily result in the smallest respectively largest area size in a mathematical sense.

In this plug-in the Bounding Box is parallel to an arbitrary, specified axis. Due to this you can receive an Object Oriented Bounding Box as well as an Axis Aligned Bounding Box, which is parallel to the image coordinate system (orientation angle = 0). See "How To Use" for further information.

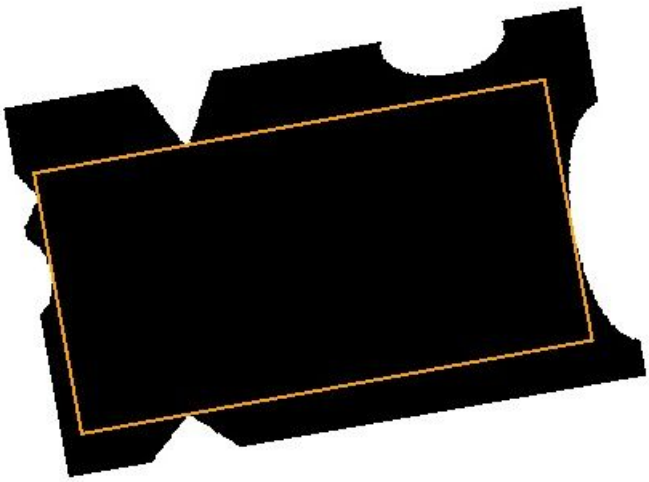
Axis Aligned Outer Bounding Box sample



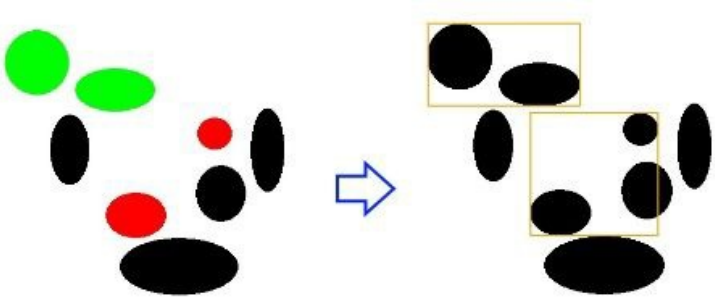
Object Oriented Outer Bounding Box sample (using principal axis from NeuroCheck feature list)



Object Oriented Inner Bounding Box sample (using principal axis from NeuroCheck feature list and raster size of 5)



Axis Aligned Outer Bounding Box around groups



Use cases

The Outer Bounding Box is widely used. For a lot of general use cases, its easiest version, the Axis Aligned Bounding Box, is sufficient. With the arbitrary axis orientation, implemented in this plug-in, further applications can be considered. For example:

- Classify different objects in an image with respect to their orientation (principal axis) on the basis of their expansion given through the Outer Bounding Box (and possibly Inner Bounding Box).
- Detect shape deviations or distortions.

Considering groups instead of single objects, examples may be:

- To determine the expansion of many small objects (e.g. pores).

The Inner Bounding Box is especially relevant in combination with the Outer Bounding Box. For example:

- To determine notches, cuts or border defects at components like solar wafers you may look at the difference between the outer and inner box.

Input data

This check function requires an image and a list of ROIs as input data objects.

Output data

Output is a list of ROIs containing the Bounding Boxes.

Result view

The result view shows the input image with the created Bounding Boxes.

Properties

 Check function group Plug-In.

 The check function has a Parameter Dialog.

Create Bounding Box: How to Use

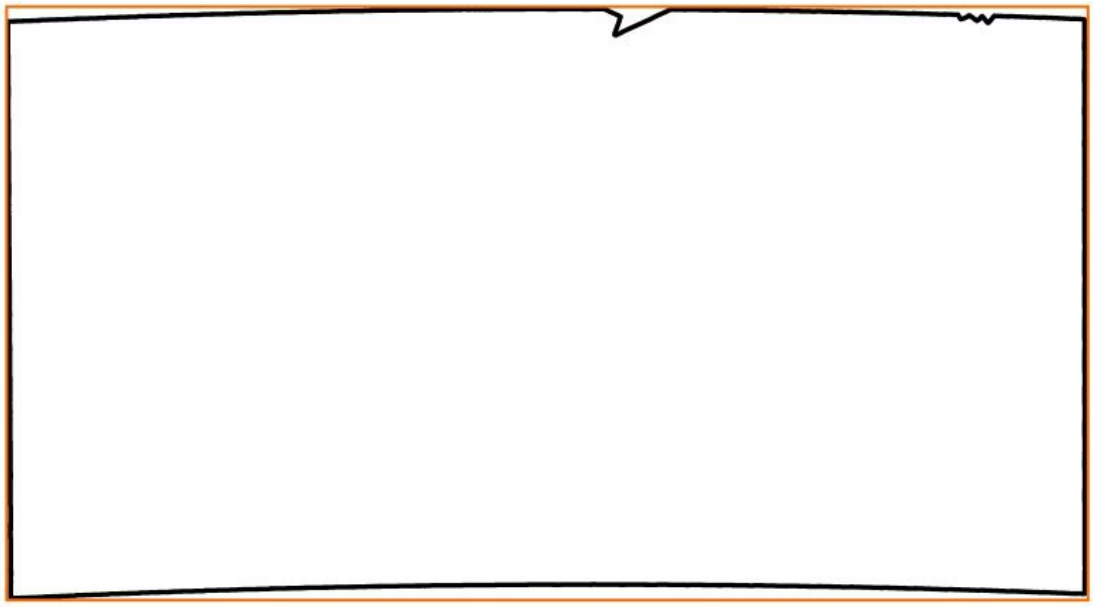
This check function calculates a Bounding Box for each object or each group in the input list of ROIs. There are two Bounding box modes available: Outer Bounding Box (OBB) and Inner Bounding Box (IBB).

Outer Bounding Box in detail

The object contour points are transformed to the new coordinate system, which consists of the given axis (e.g. principal axis from the feature list) and it's normal. Center of the coordinate system is the center of the ROI.

For the OBB simply the largest (for the top and right bound) and smallest (for the bottom and left bound) contour coordinates are searched.

Outer Bounding Box of a solar wafer sample

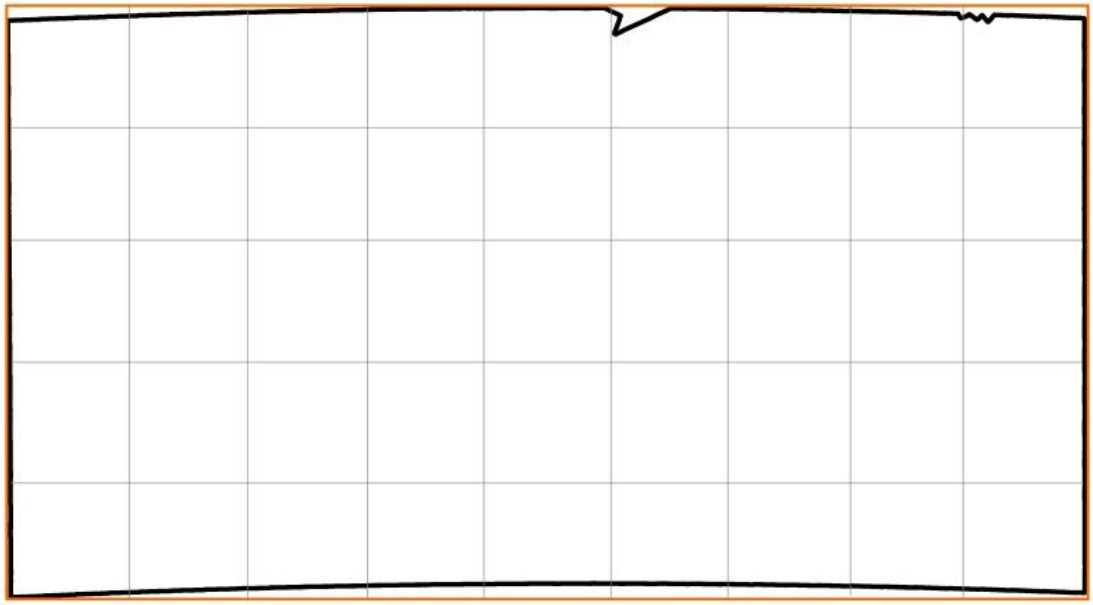


Inner Bounding Box in detail

The IBB has the OBB preceded.

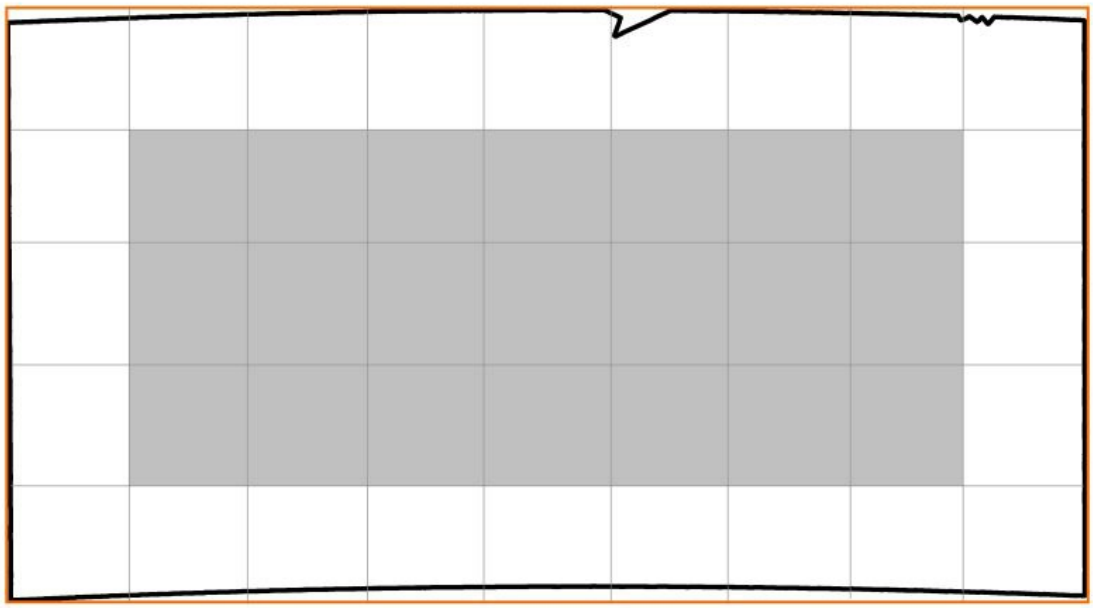
Accordingly to the user specified raster size, the area of the OBB is divided into raster cells. The raster size gives the number of raster cells along the smaller side. The raster cell dimension = (OBB side length) / (number of raster elements). As the raster cells are considered to be quadratic, the number of raster cells along the larger side = (OBB side length) / (raster cell dimension).

Wafer sample: Raster cells, having a raster size of 5

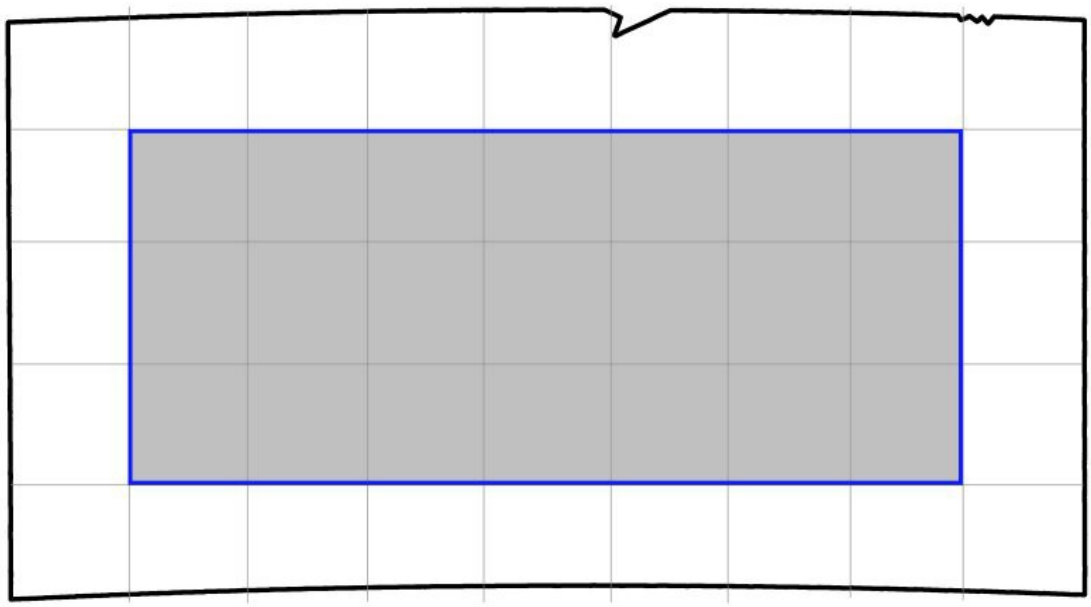


Only the raster cells lying completely inside the object are furthermore considered to find the largest rectangles, which lie completely inside the object. These rectangles serve as initial structures to find the IBB in the following extension step.

Wafer sample: Raster cells lying completely inside the object (grey)

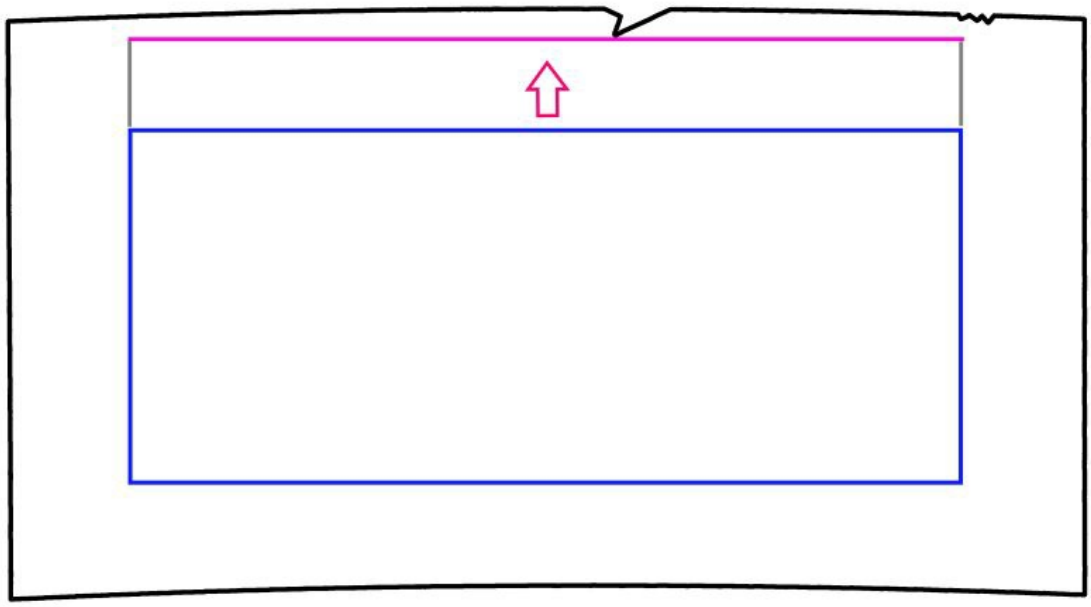


Wafer sample: Maximal Rectangle, which is found

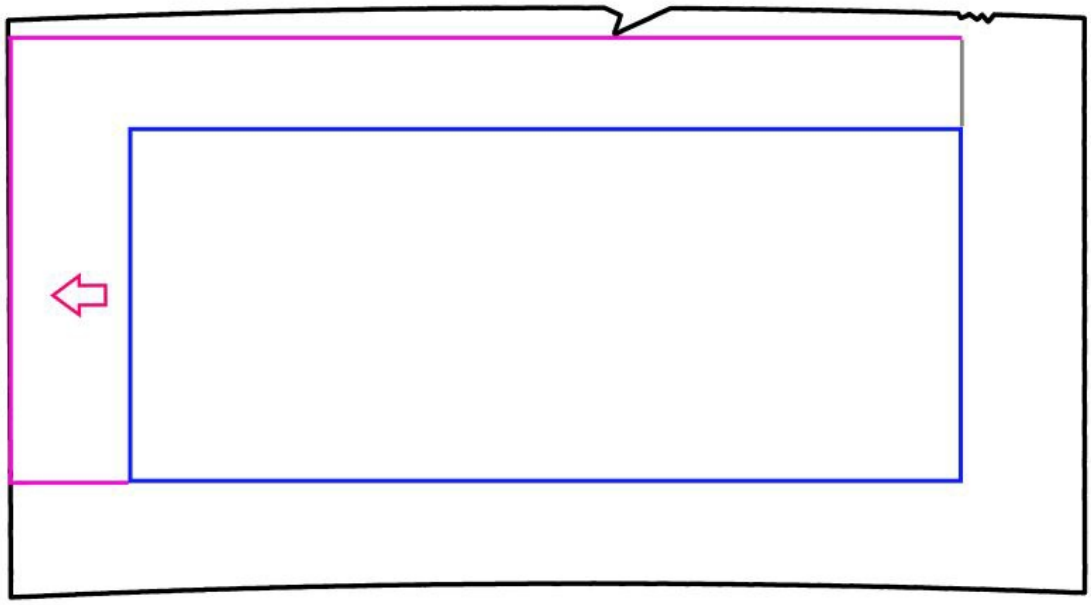


For each initial rectangle in the list its four sides are moved outwards to the smallest (top or right border) or largest point (bottom or left border) in the considered section. This extension step is running four times to have each side starting once. The resulting IBB is the one with the largest area. If there is more than one rectangle with the same area, the first one is kept as result.

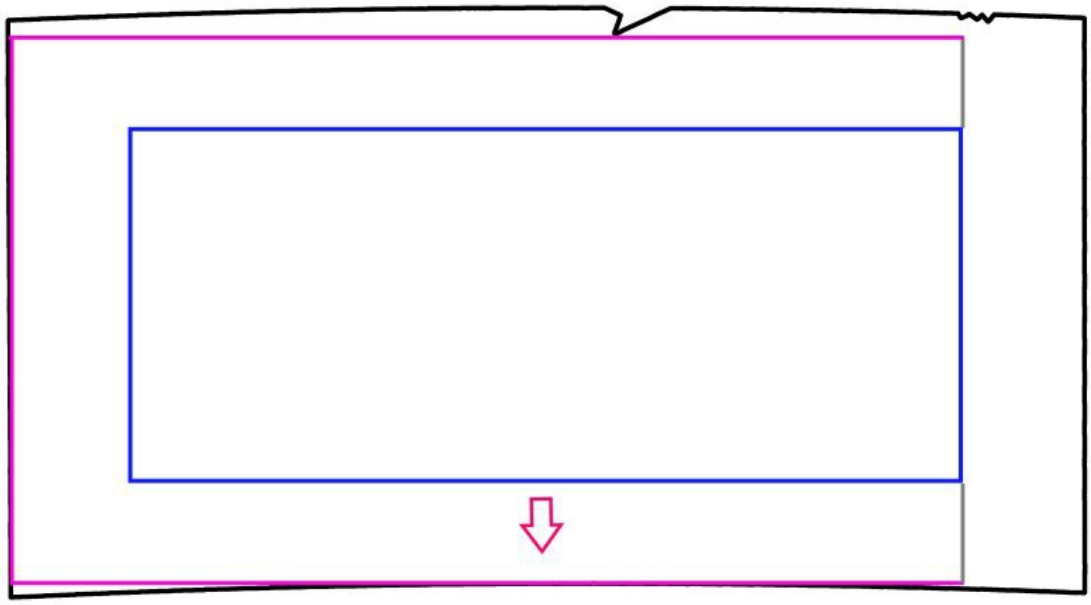
Wafer sample: Extension step to the top



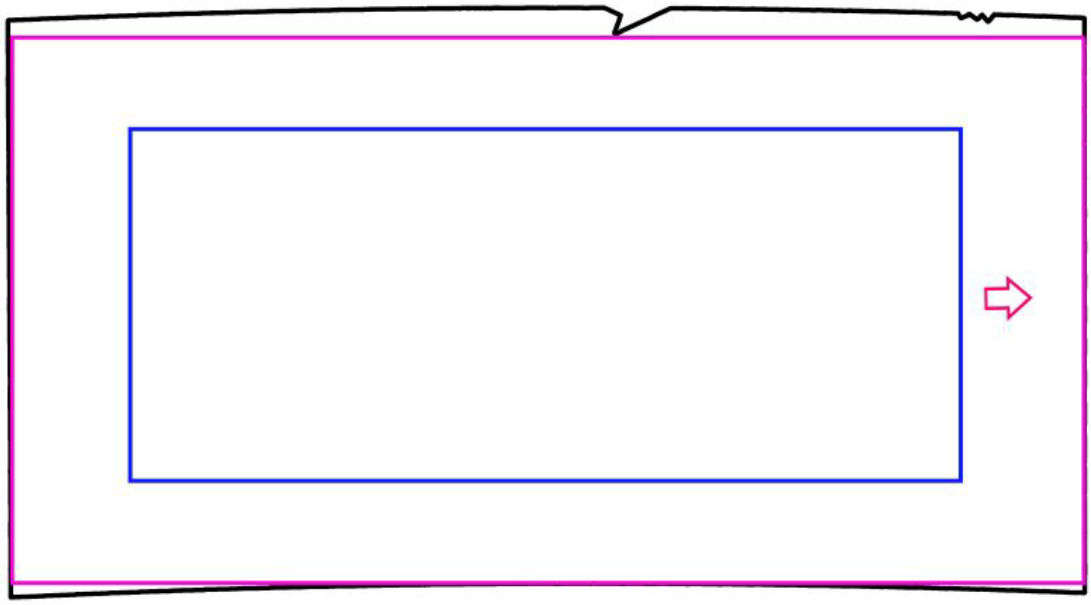
Wafer sample: Extension step to the left



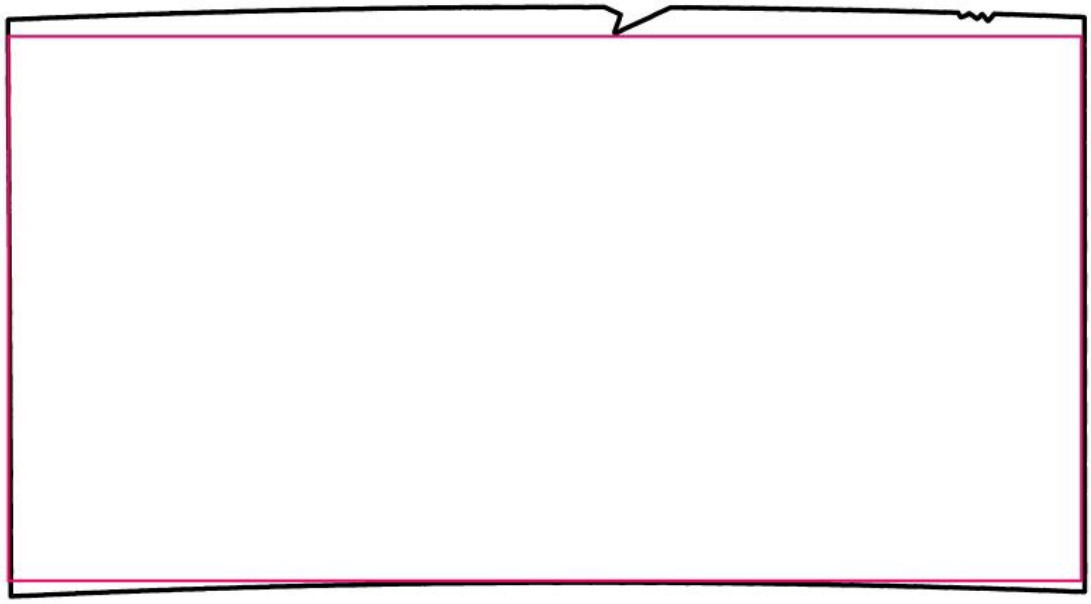
Wafer sample: Extension step to the bottom



Wafer sample: Extension step to the right



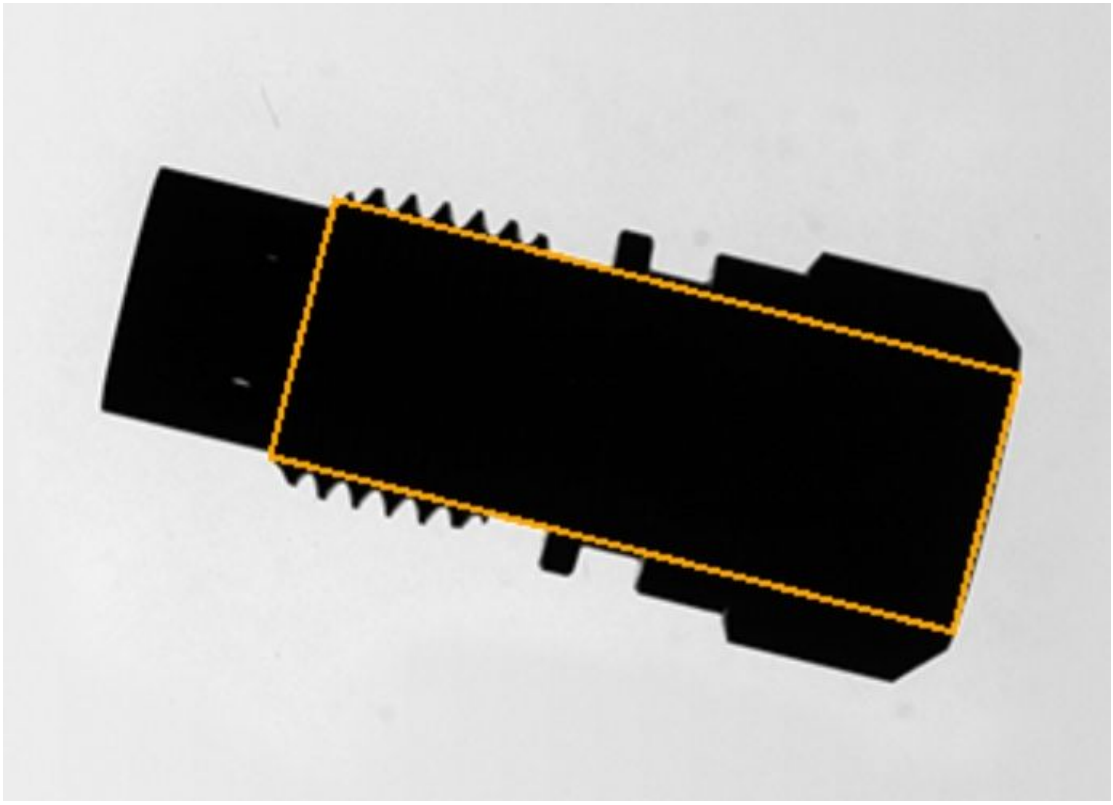
Wafer sample: Result of the Inner Bounding Box



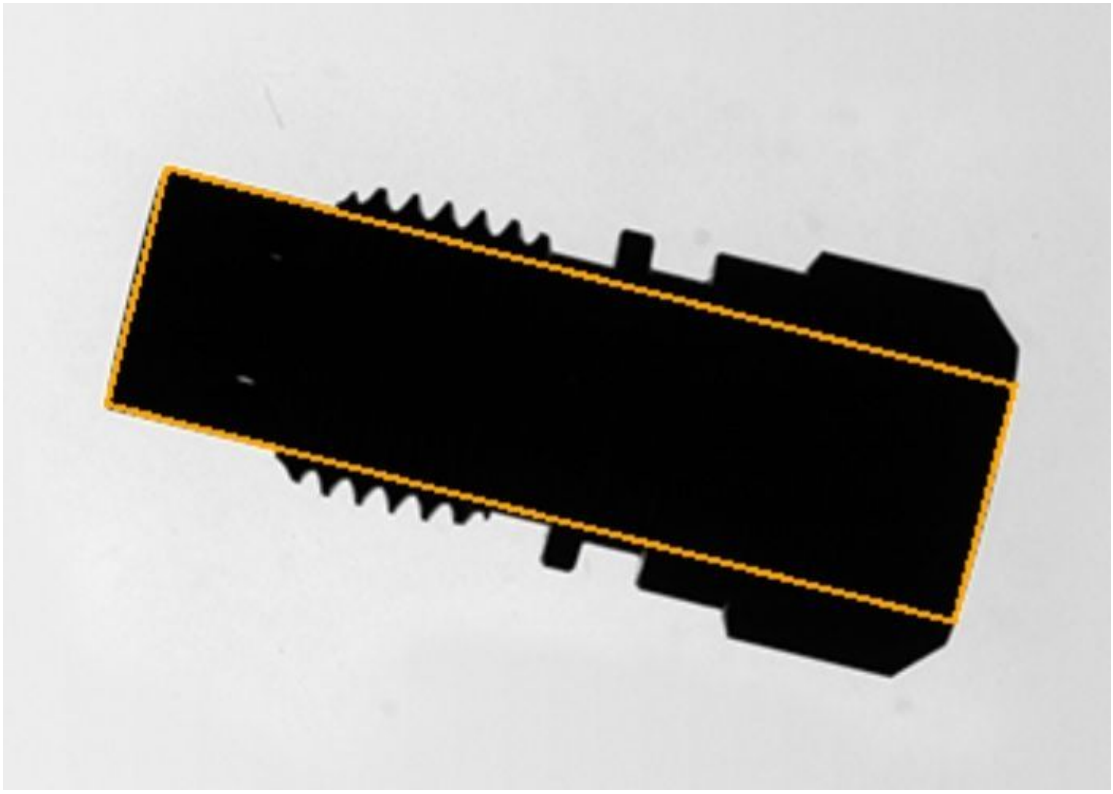
Raster size effects - Choose the parameter properly

The number of raster elements is limited from 5 to 40. If the number of raster elements is bigger than the smaller side of the Outer Bounding Box, the Inner Bounding Box can't be calculated and the result will be NIO. In general it is recommended to choose a small number, so that the performance is increased. On the other hand, it may happen, that no raster element lies completely inside the (e.g. strung-out) object and therefore the Inner Bounding Box can't be calculated, when a small number is chosen.

Sample 1 with raster size of 5 and angle from feature list: The initial rectangle position is not ideal, the extension is stopped too early



Sample 1 with raster size of 10 or bigger and angle from feature list: Correct result



Especially circular structures can cause variations of the Inner Bounding Box results when changing the raster size. Even small raster size changes can lead to a slightly different position and size of the initial rectangle. When extending the initial rectangle, it will be stopped through the circular form. Therefore the result is dependent on the initial position and size.

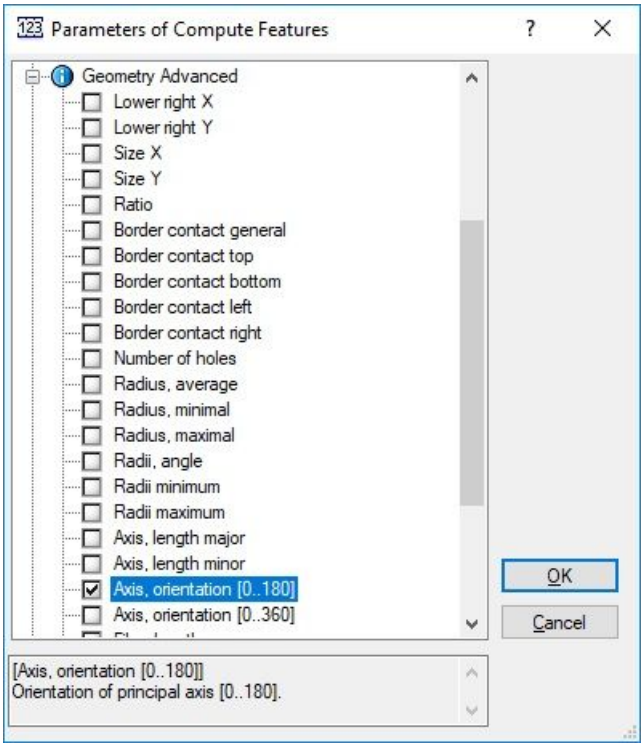
Angle source effects - Angle accuracy as important requirement

The axis orientation as angle from [0...180] can be read from register or extracted from the feature list. Considering the feature list, the principal axis orientation is used and advisable for use cases with irregular shaped objects. Keep in mind, that an approximation of the principal axis is used, which is fast but may be unstable or inaccurate for some shapes (e.g. square like, strung-out). Therefore the reliability of the principal axis orientation should be proofed beforehand for the specific use case. Otherwise it is recommended to use an angle from the register, which could be previously calculated for example as angle of an created edge to a specific object side.

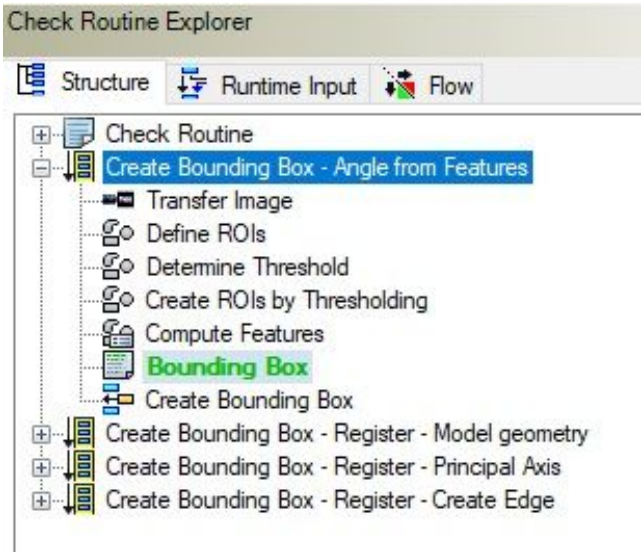
Keep in mind, that the angle accuracy has a big influence of the result. Even a small degree deviation may cause for example the IBB to stop too early.

Check routine samples

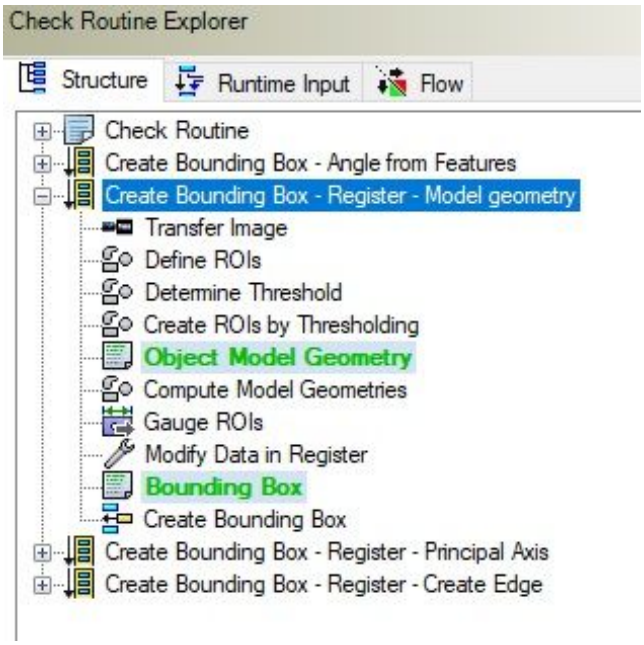
Parameter setting of Compute Features



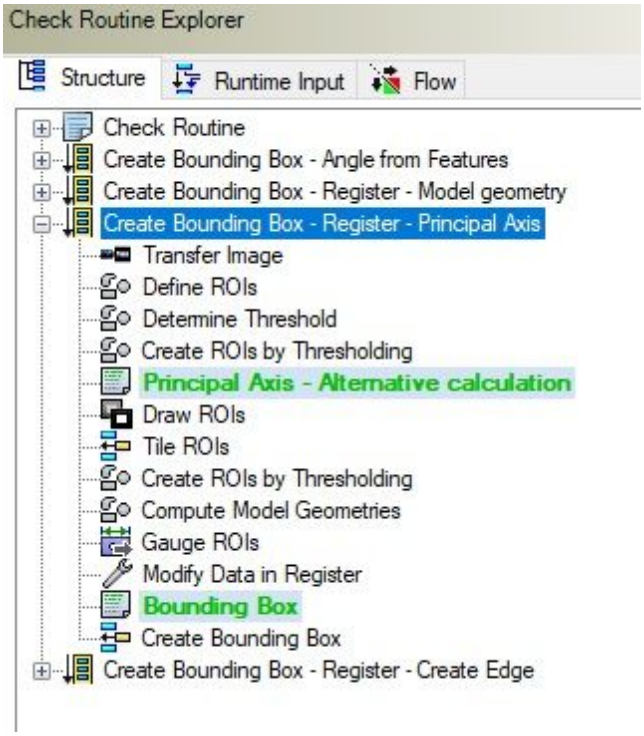
Screenshot of Check Routine Sample: Angle from features



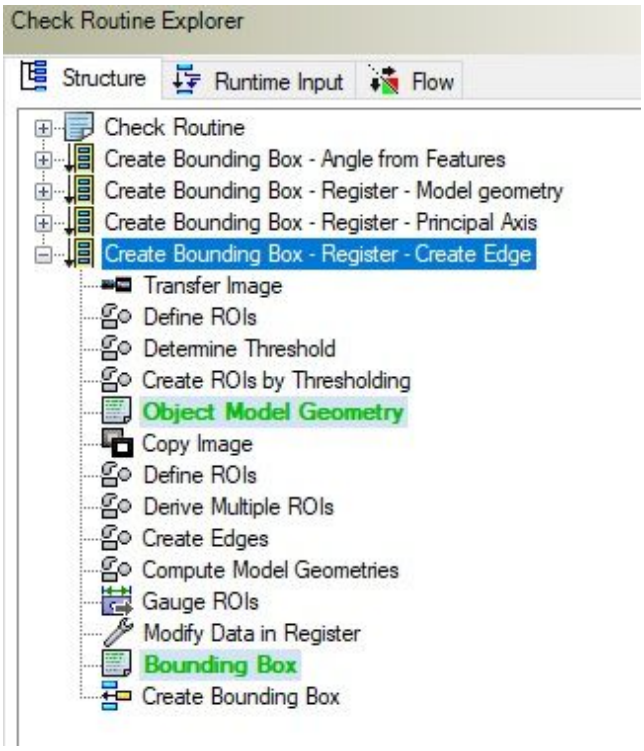
Screenshot of Check Routine Sample: Calculate angle from model geometry and write it into register



Screenshot of Check Routine Sample: Calculate principal axis angle using Tile ROIs and write it into register



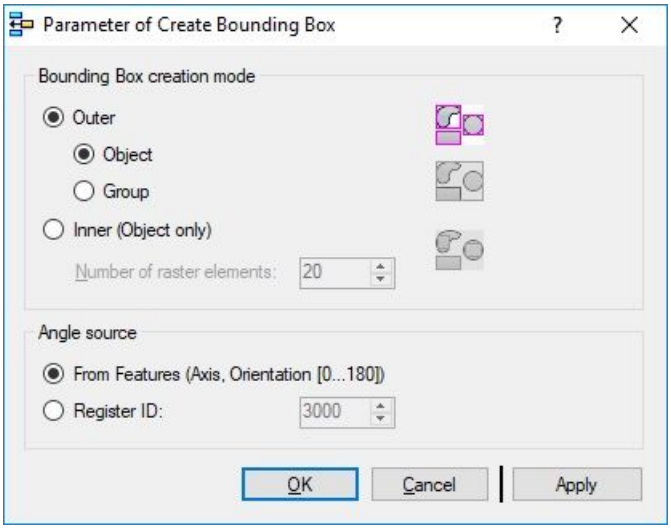
Screenshot of Check Routine Sample: Calculate angle from object edge and write it into register



Create Bounding Box: Parameter Dialog

This plug-in check function has a **Parameter** dialog.

Screenshot of Parameter Dialog



The **Parameter** dialog contains the following elements:

Element	Description
Bounding Box creation mode	There are two types of Bounding Boxes: Outer Bounding Box and Inner Bounding Box.
Outer	In the Outer creation mode, it can be selected between Object-wise or Group-wise calculation.
Object	Object-wise means an Outer Bounding Box is created for each ROI in the list.
Group	Choosing Group-wise, one Outer Bounding Box will be created to enclose all ROIs with the same group number.
Inner	In the Inner creation mode the calculation is always Object-wise.
Number of raster elements	Specifies the number of raster elements to subdivide the smaller side of the Outer Bounding Box. The number of raster elements along the other side is calculated, so that the raster elements/cells have a quadratic dimension. The number of raster elements is limited from 5 to 40. If the number of raster elements is bigger than the smaller side of the Outer Bounding Box, the Inner Bounding Box can't be calculated and the result will be NIO. In general it is recommended to choose a small number, so that the performance is increased. On the other hand, it may happen, that no raster element lies completely inside the (e.g. strung-out) object and therefore the Inner Bounding Box can't be calculated, when a small number is chosen. See "How To Use" for samples and raster size effects. Generally a number of 20 raster elements is a good parameter size to start with.
Angle Source	The angle for the Bounding Box orientation can be received either from a register or from the feature list. See "How To Use" for samples and angle source effects.
From Features (Axis, Orientation [0...180])	The principal axis from the feature list can be used, if an Object-wise calculation is selected. The feature 'Axis, Orientation [0...180]' has to be calculated beforehand, applying the check function 'Compute Features'.
Register ID	Specifies the source register for the orientation angle.

Create Positioned ROIs: Introduction

Function overview

This check function can be used to create new ROIs in relation to an existing input ROI.

For each input object, its reference point builds the start for calculating the new geometry object. Possible reference points for an input object can be its center point, or the top left, top right, bottom left or bottom right corner of its enclosing rectangle. The offset shifts the reference point in X and Y direction. That reference point is used as new start coordinate if the new line, the center coordinate of the new circle or the top-left coordinate of a new AOI.

Input data

This check function requires an image and a list of ROIs as input data objects.

Output data

None (changed list of ROIs)

Result view

The result view shows the output ROIs.

Properties

 Check function group Plug-In.

 The check function has a Parameter Dialog.

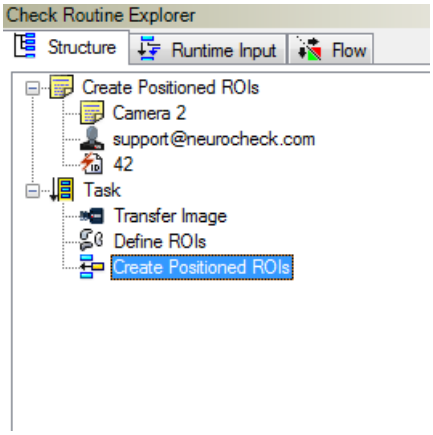
 The check function has own result Visualizations.

Create Positioned ROIs: How to Use

This check function can be used to create new ROIs in relation to an existing input ROI.

Check routine sample

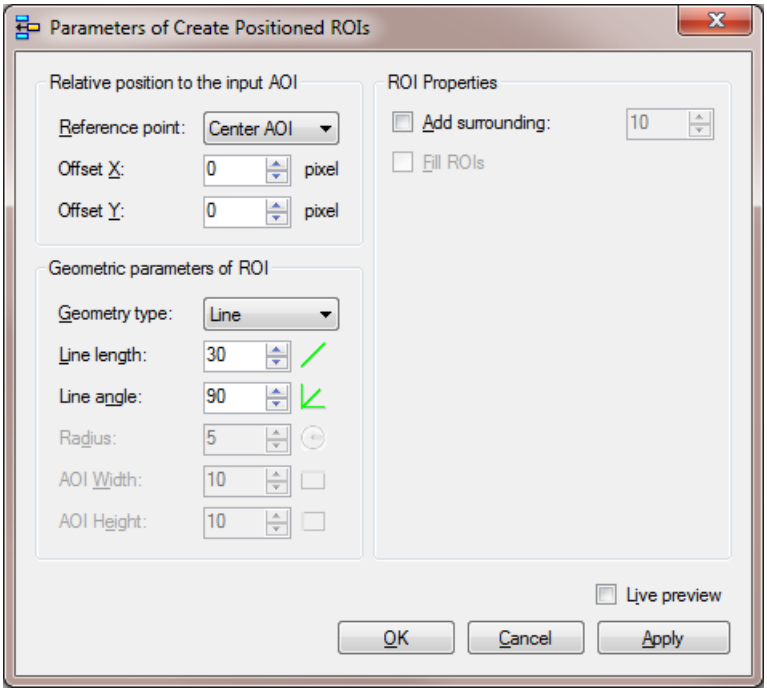
Screenshot of Check Routine Example



Create Positioned ROIs: Parameter Dialog

This plug-in check function has a **Parameter** dialog.

Screenshot of Parameter Dialog



The **Parameter** dialog contains the following elements:

Element	Description
Reference point	This reference point defines the start position of the new ROI.
Offset X / Y	It is possible to add an offset to the new starting position.
Add surrounding	To add a surrounding area to a line or a circle, check this option and enter the width in pixels. The surrounding area must not leave the image. This option is only applied to new ROIs converted from a model geometry.
Fill ROIs	With this option you can fill a surrounding area of a line or a circle, if this area is created by "Add surrounding area". Circles without surrounding area can be filled, too. This filling is only applied to new ROIs converted from a model geometry.
Geometry type	Defines the geometry type of the new ROI.
Line length	Defines the length of a line. Only active if geometry type line is selected.
Line angel	Defines the angle of a line. Only active if geometry type line is selected.
Radius	Defines the radius of a circle. Only active if geometry type cicle is selected.
AOI Width	Defines the AOI width of a rectangle. Only active if geometry type AOI is selected.

Element	Description
AOI Height	Defines the AOI height of a rectangle. Only active if geometry type AOI is selected.
Add surrounding	Width of the surrounding area for each line or cicle.
Fill ROIs	If activated, the surrounding area or the cicle itself will be filled.

Create Positioned ROIs: Visualizations

This section describes the result visualizations the check function "Create Positioned ROIs" provides.

Element	Description
Output ROIs	Shows the new and changed ROI collection.

Create ROIs from Model Geometries: Introduction

Function overview

This function converts input ROIs into a list of output ROIs without model geometries.

Input data

This check function requires a list of ROIs as input data object.

Output data

New list of ROIs.

Result view

The result view shows the new output ROIs.

Properties

Check function group Plug-In.

The check function has a Parameter Dialog.

The check function has own result Visualizations.

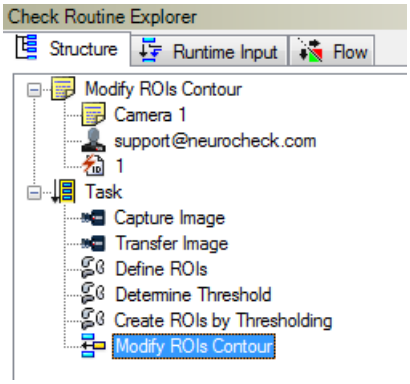
Create ROIs from Model Geometries: How to Use

This function converts input ROIs into a list of output ROIs without model geometries. Furthermore a surrounding area can be specified which can also be filled.

Please note, that feature values from an original ROI with a model geometry are not copied to the new ROI. Copied ROIs will keep their feature values.

Check routine sample

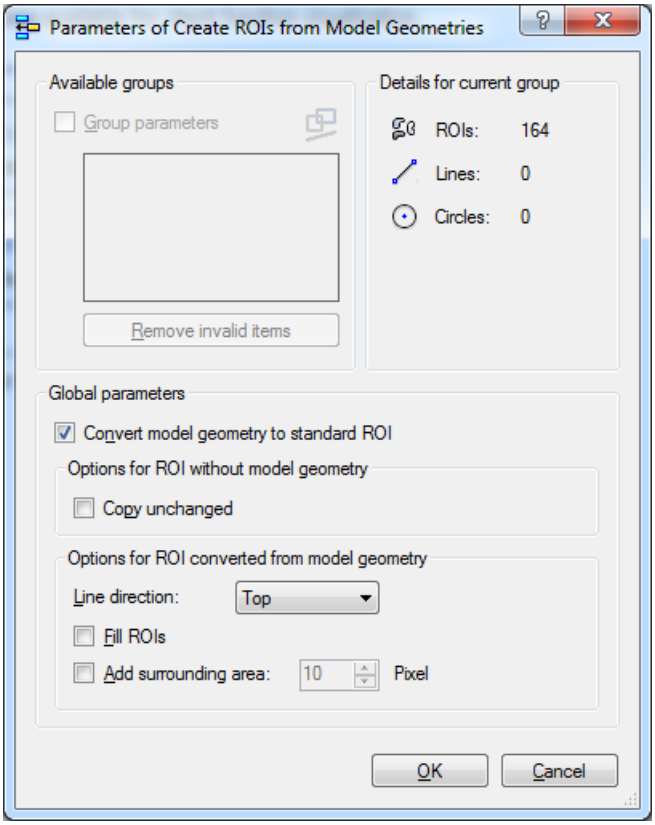
Screenshot of Check Routine Sample



Create ROIs from Model Geometries: Parameter Dialog

This plug-in check function has a **Parameter** dialog.

Screenshot of Parameter Dialog



The **Parameter** dialog contains the following elements:

Element	Description
Group parameters	Group mode is only available if the grouping properties have been activated in the check function "Define ROIs". If deactivated parameter settings are treated globally, i.e. all settings of the parameter group are the same for all groups. If activated the box containing ROI groups with group numbers is shown below the check box. Parameter settings are treated groupwise.
Convert model geometry to standard ROI	Activate here the calculation. If more than one group is used the calculation is activated for each group explicit.
Copy unchanged	If an input ROI does not have a geometry model then this checkbox can be activated to copy this ROI without changes.
Line direction	Because a model geometry does not have direction information in 360 degrees but the line object has you can specify here which end of the line will be the starting point.
Fill ROIs	With this option you can fill a surrounding area of a line or a circle, if this area is created by "Add surrounding area". Circles without surrounding area can be filled, too. This filling is only applied to new ROIs converted from a model geometry.

Element	Description
Add surrounding area	To add a surrounding area to a line or a circle, check this option and enter the width in pixels. The surrounding area must not leave the image. This option is only applied to new ROIs converted from a model geometry.

Create ROIs from Model Geometries: Visualizations

This section describes the result visualizations the check function "Create ROIs from Model Geometries" provides.

Element	Description
Output ROIs	Shows the output ROIs of the check function.

Load ROIs: Introduction

Function overview

This Plug-In function loads ROIs from an XML document. It can be used to restore a list of ROIs formerly saved by the check function Save ROIs. As the loading source the user may specify:

- File with a fixed path;
- File with a path taken dynamically from the data register;
- Data register of type String to load the document from an XML string.

Input data

This check function requires an image as input data object.

Output data

Loaded list of ROIs.

Result view

The result view shows the input image and the loaded output ROIs.

Properties

 Check function group Plug-In

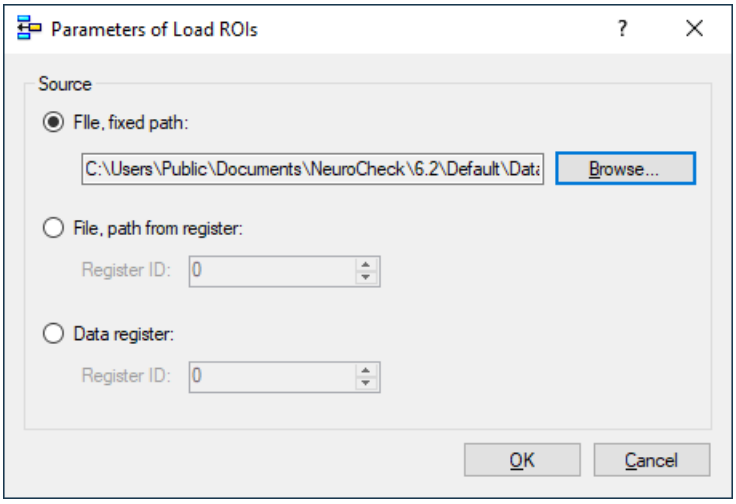
 The check function has a Parameter Dialog.

Load ROIs: Parameter Dialog

Parameter

This plug-in check function has a **Parameter** dialog.

Screenshot of Parameter Dialog



The **Parameter** dialog contains the following elements to specify the source:

Element	Description
File, fixed path	Select an existing file path.
File, path from register	Select a register cell containing a file path as String.
Data register	Select a register cell containing the document as an XML string

Load ROIs from Directory: Introduction

Function overview

This check function loads ROIs from a list of XML documents, one document at a time. It can be used to restore a list of ROIs formerly saved by the check function Save ROIs. As the loading source the user may specify:

- A fixed path;
- A path taken dynamically from the data register;

Input data

This check function requires an image as input data object.

Output data

The check function returns the current ROI-list as output data object.

Result view

The result view shows the input image and the currently loaded output ROIs.

Properties

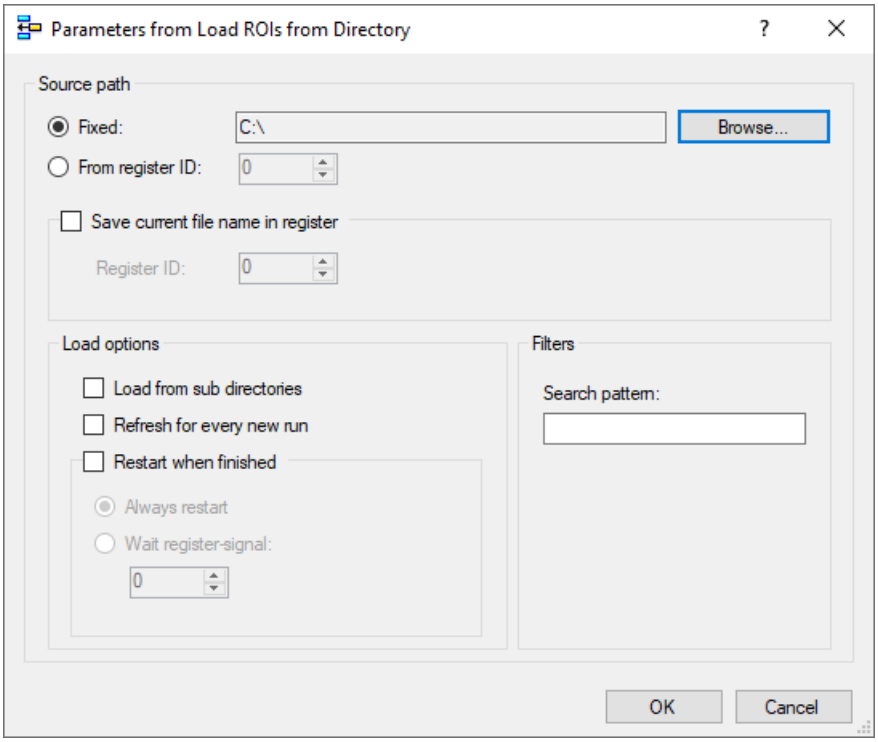
Check function group Plug-In.

The check function has a Parameter Dialog.

Load ROIs from Directory: Parameter Dialog

This plug-in check function has a **Parameter** dialog.

Screenshot of Parameter Dialog



The **Parameter** dialog contains the following elements:

Element	Description
Source path	Choose the source of the file name from the following options: • **Fixed:** Loads the ROIs from a fixed directory. • **From register ID:** Gets the path name from a register cell and then loads the ROIs. The register cell has to have the data type String.
Save current path in register	Saves the filename of the current ROI-list to a specified register cell: • **Register ID:** The ID of the register cell, where the path of the current ROI list is saved. The register cell has to have the data type String.
Filters	Apply a set of filters when listing a directory: • **Search pattern:** The search string to match against the names of files in path. This parameter can contain a combination of valid literal path and wildcard (* and ?) characters, but it doesn't support regular expressions. This applies only to the files listed and not the path name itself. ◦ More info at: https://docs.microsoft.com/en-us/dotnet/api/system.io.directory.enumeratefiles?view=net-5.0

Element	Description

Load options	Indicates how the ROIs should be listed:
	• **Load from sub directories:** Includes sub directories in the listing process.
	• **Refresh for every new run:** Reloads the list of files every time the check function is executed.
	• **Restart when finished:** Restart from the beginning after all ROIs in the directory have been listed:
	  ○ **Always restart:** Always restart when list-end is reached.
	  ○ **Wait register-signal:** Restart only if the specified register cell contains a Boolean value of "True". After restarting, the register cell is set back to "False".

The "Search pattern" gives the opportunity to filter items based on indexes, dates or other partial text patterns. Examples:

1. The search pattern **2021*** will make the check function list only files that start with the prefix **2021**.
2. ***123.xml** will list all XML files that end with the suffix **123**.
3. ***abc*cde.xml** will list all XML files with **abc** in the middle and that end with **cde**.

Use the parameter "Refresh for every new run" when new files might be created or copied to the specified directory after the list is initialized. However, it can happen that the new files are placed before the current item in the list, since the check function lists files by name (alphabetical order). In these cases, the new files will be skept but will be included in the next run if the "Restart when finished" is active.

Merge ROIs: Introduction

Function overview

This check function creates ROIs by applying a specific logical operation to regions of the input ROIs. All ROIs are converted to binary regions, the operation is applied to all regions, and new ROIs are created from each independent region.

Supported operations are:

- **Union:** unites all overlapping or contacting regions
- **Intersection:** leaves overlapping (common) regions of all ROIs only
- **Intersection over groups:** leaves overlapping (common) regions of all ROI groups only
- **Inversion:** computes the background of all regions

Use cases

- Multiple closely located or overlapping ROIs should be merged into one object
- Only an overlapping area of input ROIs or input ROI groups is required
- The background of all ROIs has to be segmented

Input data

This check function requires an image and a list of ROIs as input data objects.

Output data

New list of ROIs after applying an operation.

Result view

The result view shows the new output ROIs, the input ROIs and the input image.

Properties

 Check function group Plug-In.

 The check function has a Parameter Dialog.

 The check function has own result Visualizations.

Merge ROIs: How to Use

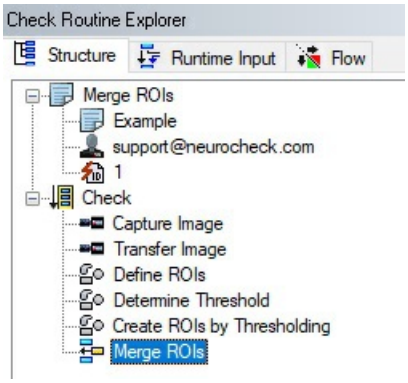
There are three different merging operations available. It is possible to unite, intersect, or invert input regions.

All operations work on binary regions only, so all input objects are converted to binary regions. Output ROIs are binary objects as well.

After applying the operation each independent region, which is determined by pixelwise (8-neighborhood) connection tracing, is converted to a single ROI.

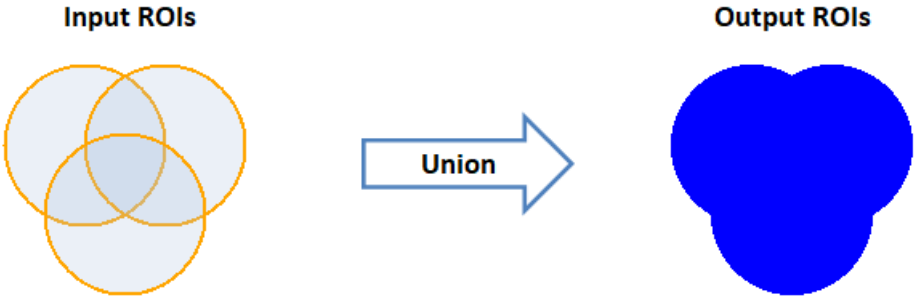
Check routine sample

Screenshot of Check Routine Sample



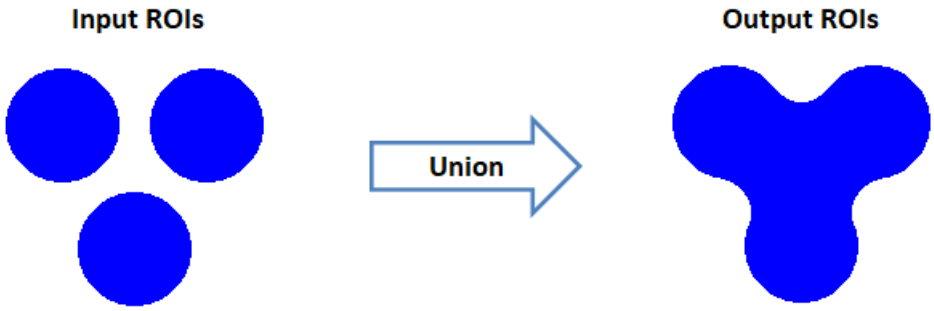
Merging operations

- **Union:** All ROI regions are combined and those of them that overlap or contact each other build up new independent objects.

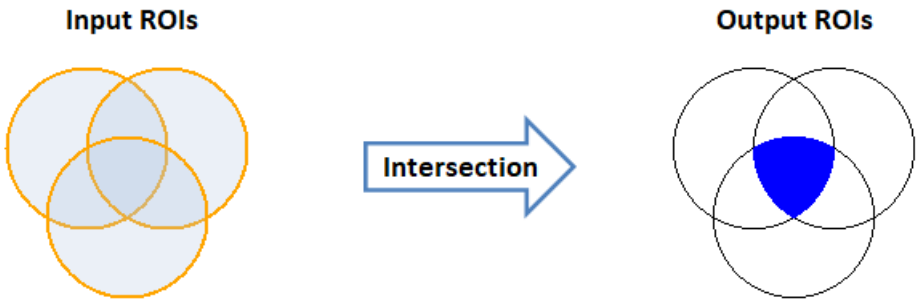


Closing: Regions can also be filtered with the closing filter, that allows to connect closely located regions and smooth their contours. Overlapping regions are identified after filtering.

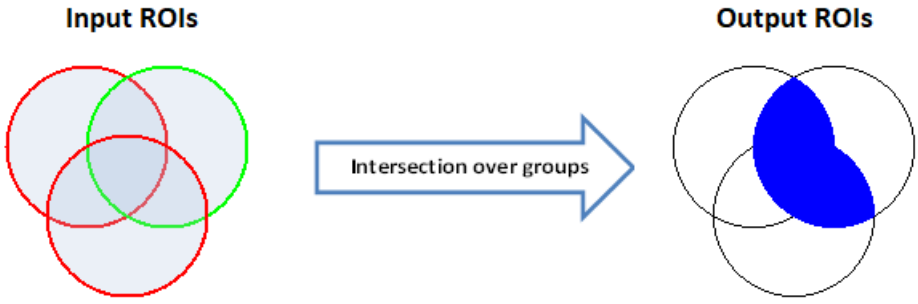
Sample of Union with Closing



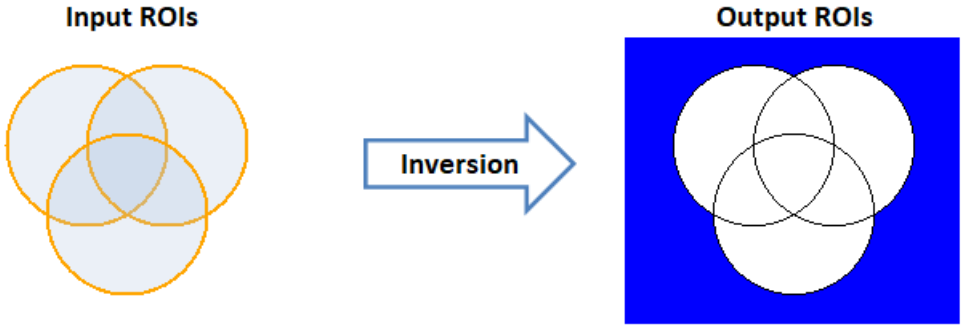
- **Intersection:** All regions from each ROIs are intersected with each other until only those areas left, which are present in all ROIs. Intersection on a single ROI produces regions from this ROI.



- **Intersection over groups:** All regions from each group of ROIs are intersected with each other until only those areas left, which are present in all input ROI groups. Intersection on a single group produces a union over all ROI regions in the group.



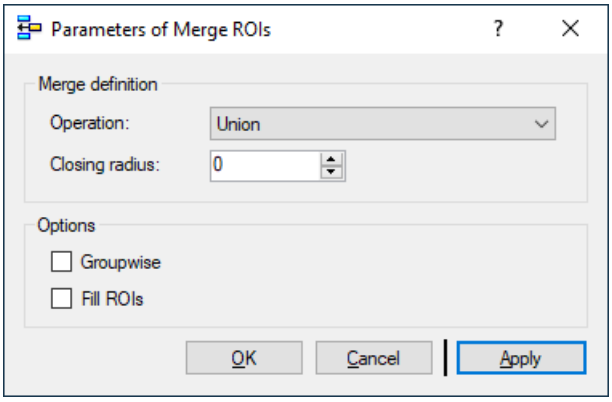
- **Inversion:** All ROI regions are combined and then inverted, producing only those regions, which do not belong to any ROI (background). Output regions are restricted by the input image bounds.



Merge ROIs: Parameter Dialog

This plug-in check function has a **Parameter** dialog.

Screenshot of Parameter Dialog



The **Parameter** dialog contains the following elements:

Element	Description
Operation	Defines the merging operation: **Union** Creates new ROIs from the overlapping or contacting regions of all ROIs. **Intersection** Creates new ROIs from the overlapping region parts of all ROIs only. **Intersection over groups** Creates new ROIs from the overlapping region parts of all ROI groups only. **Inversion** Creates new ROIs by inverting regions of all ROIs. Output ROIs are restricted by the image bounds.
Closing radius	Defines the kernel size of the morphological closing filter, that is applied to regions of ROIs during the union operation.
Groupwise	Defines whether the input ROIs from different groups should be processed separately. Output ROIs get assigned to the corresponding groups.
Fill ROIs	Fills newly created ROIs. Filling does not change the number of independent ROI objects in the output.

Merge ROIs: Visualizations

This section describes the result visualizations the check function "Merge ROIs" provides.

Element	Description
Output ROIs	Shows the output ROIs of the check function.
Input ROIs	Shows the input ROIs of the check function.
Input Image	Shows the input image of the check function.

Modify ROIs Contour: Introduction

Function overview

This check function modifies ROIs contour. It is possible to grow, shrink or create a surrounding of all incoming ROIs.

Use cases

- Additional area should be added to the input ROIs.
- Area around the contour of the input ROIs should be ignored.
- Only a surrounding of the input ROIs contour should be analyzed e.g. looking for a splint.

Input data

This check function requires an image and a list of ROIs as input data objects.

Output data

Modified list of ROIs.

Result view

The result view shows the modified output ROIs, the input ROIs and the input image.

Properties

 Check function group Plug-In.

 The check function has a Parameter Dialog.

 The check function has own result Visualizations.

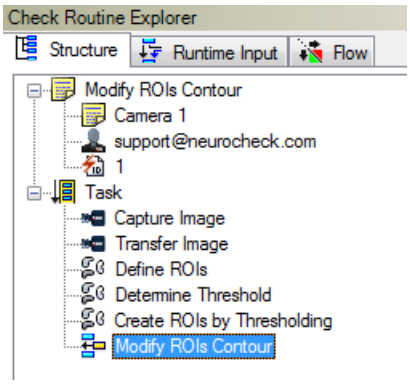
Modify ROIs Contour: How to Use

This check function can be used to modify ROIs' contour.

There are three different modification methods available. All values are given in pixel. It is possible to grow, shrink or create a surrounding of input ROIs contour.

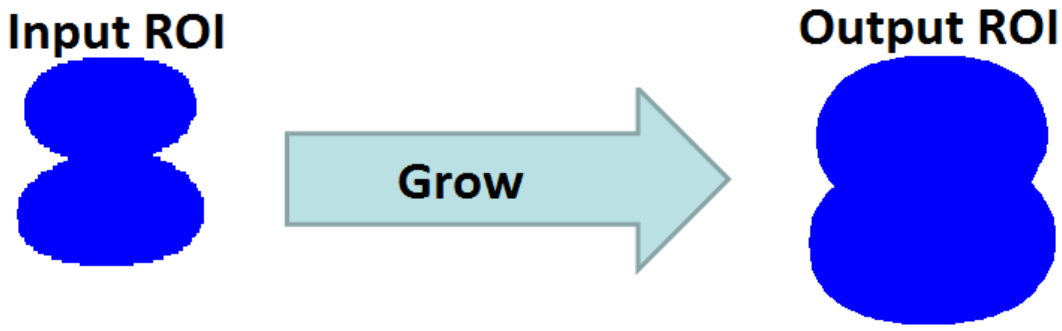
Check routine sample

Screenshot of Check Routine Sample



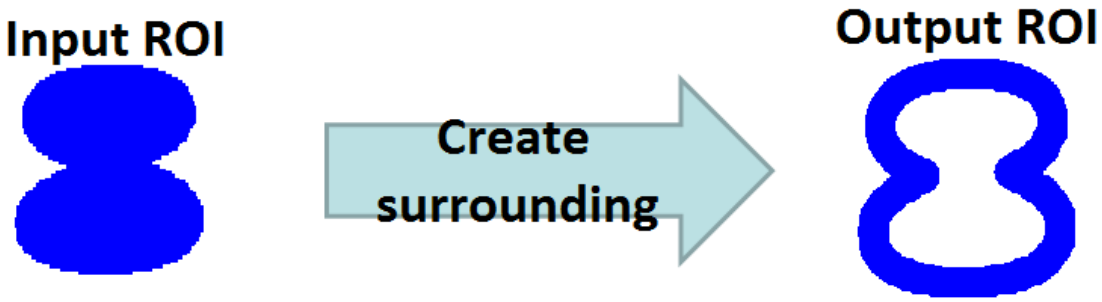
Modification methods

- **Grow**
  Lets the contour of ROIs grow. Contours are extended in all directions. If ROI has surrounding, only the surrounding is extended. Otherwise if ROIs contour is not closed, e.g. lines, arcs or polygons, then the surrounding is added. For the binary objects the modification is similar to the morphological dilation applied to the input ROIs region.
  Sample of Grow Method



- **Shrink**
  Lets the contour of ROIs shrink. Contours are reduced in all directions. If ROI has surrounding, only the surrounding is reduced. For the binary objects the modification is similar to the morphological erosion applied to the input ROIs regions. Note that filled ROIs and ROIs with closed contour may disappear, if the reduction is too large. There might also be multiple output objects created after eroding a single ROI.
  Sample of Shrink Method

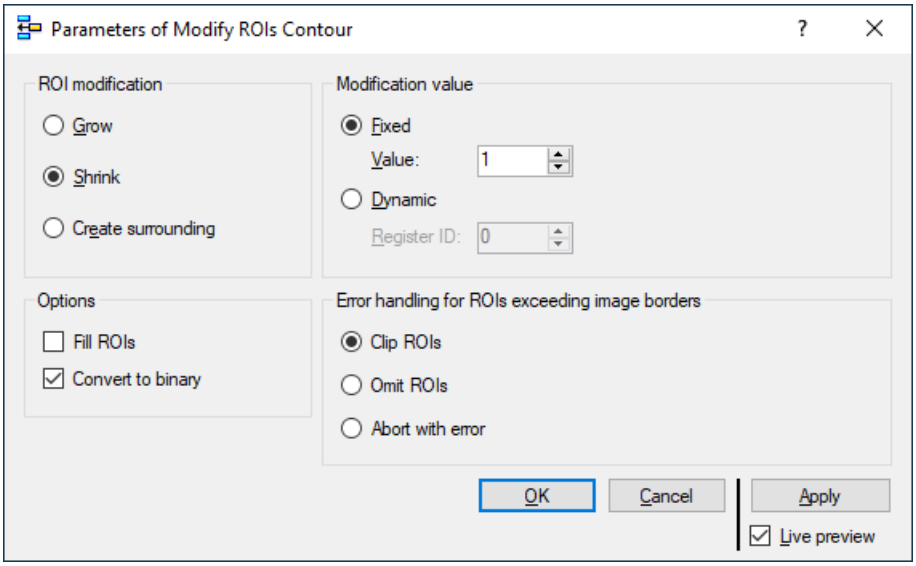
- **Create surrounding**
  Creates surrounding around the outer ROIs contours. If ROI already has surrounding, it will be replaced.
  Sample of Create surrounding Method



Modify ROIs Contour: Parameter Dialog

This plug-in check function has a **Parameter** dialog.

Screenshot of Parameter Dialog



The **Parameter** dialog contains the following elements:

Element	Description
Grow	Grows the contour of ROIs. The size of growing is given in pixel.
Shrink	Shrinks the contour of ROIs. The size of shrinking is given in pixel.
Create Surrounding	Creates surrounding around the outer ROIs contours. The surrounding size of the contour is given in pixel. The surrounding is replaced, if the ROI already had surrounding. The surrounding is removed if the modification value equals 0.
Fixed value	Defines a fixed value for the ROI modification.
Dynamic value	Reads the value from the register ID for the ROI modification. The register cell must be of type "Integer".
Fill ROIs	Activates filling for all ROIs after modification except for ROIs, that cannot be filled (such as arcs and lines without surrounding).
Convert to binary	Converts all input object to binary objects before the modification. This option also allows to clip ROIs exceeding image borders.
Error handling for ROIs exceeding image borders	Defines the function behavior, when some ROIs exceed image borders after modification: **Clip ROIs** Cuts the output ROIs to the image borders. Only binary objects are supported. **Omit ROIs exceeding image borders** Skips such ROIs and continues the execution. **Abort with error** Aborts execution when some ROI exceeds image borders.

Modify ROIs Contour: Visualizations

This section describes the result visualizations the check function "Modify ROIs Contour" provides.

Element	Description
Output ROIs	Shows the output ROIs of the check function.
Input ROIs	Shows the input ROIs of the check function.
Input Image	Shows the input image of the check function.

Reverse Transform Unrolled ROIs: Introduction

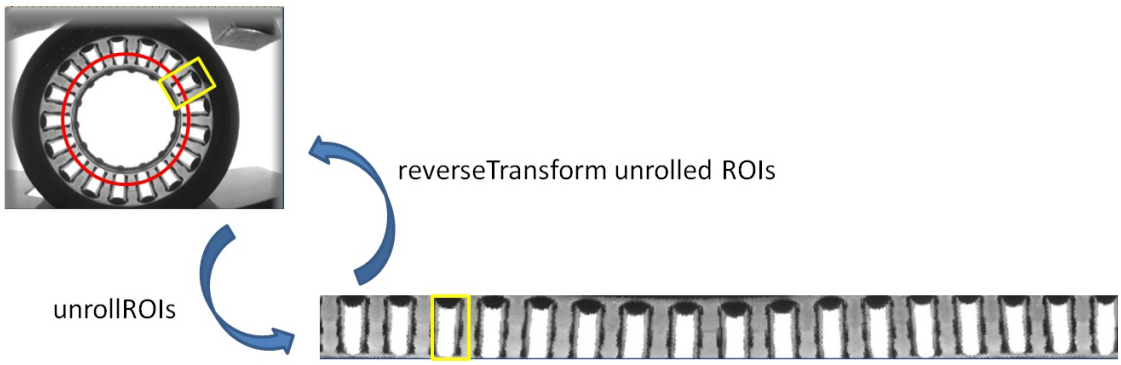
Function overview

This Plug-in check function can be used to transform ROIs, detected in an unrolled image into the original image.

The outstanding of this check function is that it has dependencies to the standard check function "Unroll ROIs". It is essential to consider carefully the introductions, defined in How to Use in order to be able to apply this check function properly.

Use cases

The main use case of this plug-in check function is illustrated in the following example. The left image is defined as the original image, it is used together with the red marked ROI as input parameter for the standard check function "Unroll ROIs". The output of this operation is the unrolled image, illustrated at the right. After unrolling the original image the yellow marked ROI is detected in the unrolled image (with the help of a ROI creation method, like Template Matching). To transform this ROI into the original image, this plug-in check function can be used. The output of this check function is the reverse transformed ROI, marked in yellow, in the original image to the left.



Restrictions

This plug-in check function has some restrictions. It does not deliver a result if one of the following parameter is enabled in the standard check function "Unroll ROIs", that creates the unrolled image:

1. Fixed length or Fixed width
2. Sampling (deviation from the default value "20")
3. Smooth Contour
4. Unroll all ROIs

Input data

This check function requires:

1. An image
2. A list of ROIs
3. An (unrolled) image, which was created by unrolling the first input-image.
4. A list of ROIs in the unrolled image.

Output data

List of reverse transformed ROIs, in the original image.

Result view

The result view shows:

1. The original image with the reverse transformed ROIs

2. The original image with the original ROIs
3. The unrolled image with the unrolled ROIs.

Properties

 Check function group Plug-In.

 The check function has a Parameter Dialog.

 The check function has own result Visualizations.

Reverse Transform Unrolled ROIs: How to Use

This Plug-in check function can be used to transform ROIs, from an unrolled image into the original image. The outstanding of this check function is that it has dependencies to the standard check function "Unroll ROIs".

Instruction:

There are three steps that have to be considered when applying this plugin-check function:

1. Execute standard check function "Unroll ROIs" first:

It is essential to first apply the standard check function "Unroll ROIs" in the check routine sequence, in order to generate the unrolled image as Input data set for this check function.

2. Hand over the parameters from "Unroll ROIs" to "Reverse Transform Unrolled ROIs" properly:

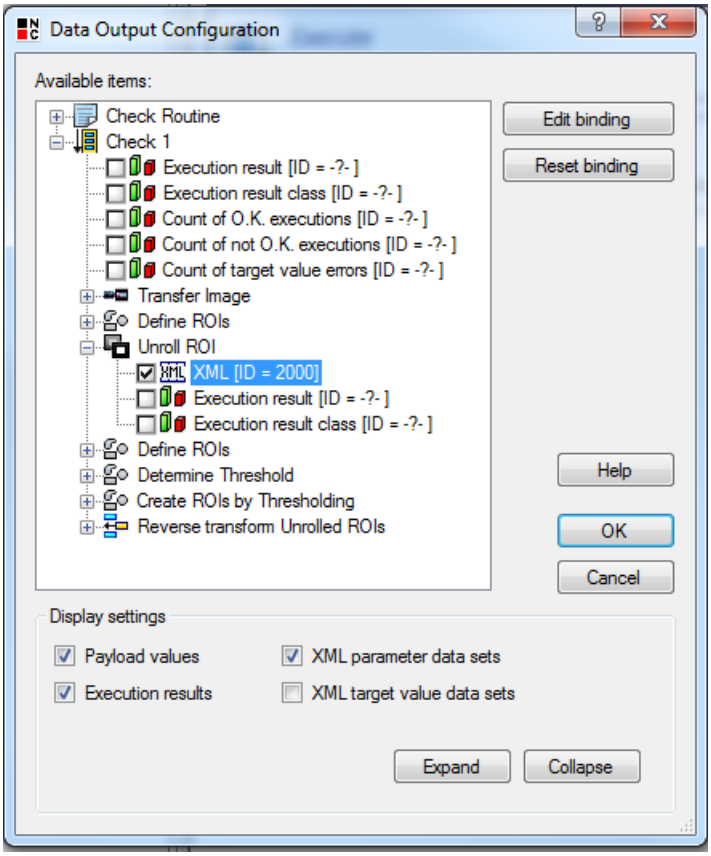
This plug-in check function needs the parameter configuration of the standard check function "Unroll ROIs" as input. To provide this information, the XML parameter data set of "Unroll ROIs" has to be bind to a data output register, by executing the following steps:

1. Create a data output register of data type "String".

2. Open the Data Output Configuration dialog. Enable the check box "XML parameter data sets", as the binding of XML parameter data sets is disabled by default.

3. Bind the XML parameter data set from the standard check function "Unroll ROIs" to a data register by selecting the check box "XML" of the check function "Unroll ROIs". The opening dialog presents the possible data registers to which the XML parameter data set can be bound.

The following screen shot shows a successful binding of the XML parameter data set to the data register with the ID 2000.



4. Make sure that Menu Tools "Use data register in manual mode" is activated.

5. Define the bound register ID in the parameter dialog of the plug-in check function "Reverse Transform Unrolled ROIs".

3. Define Runtime Input data for "Unroll ROIs" correctly:

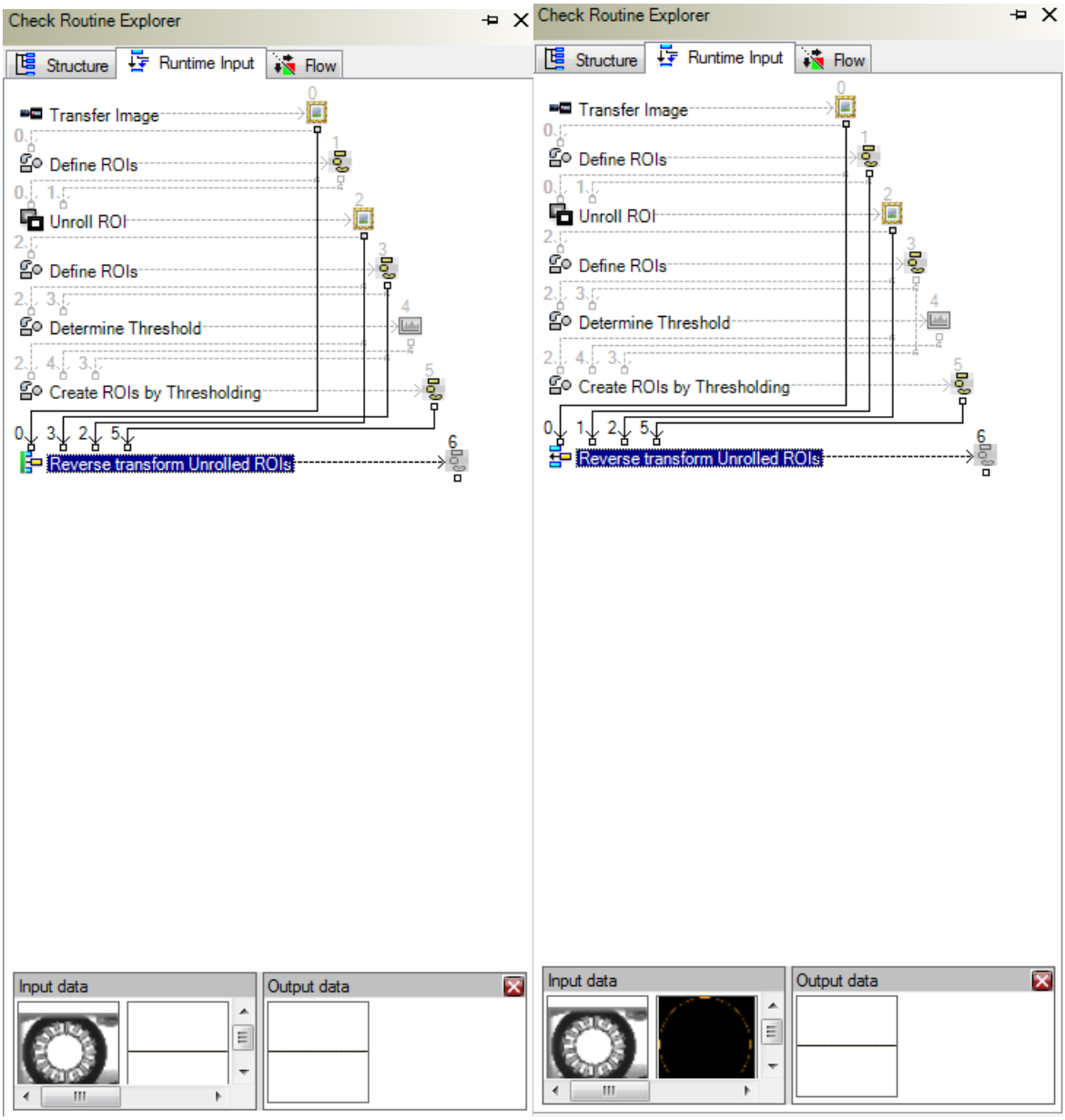
This plug-in check function has four data inputs: The original image, the ROI which was used for the unroll process, the unrolled image and the ROIs in the unrolled image. It is essential to define these data sets correctly and to respect the order they are handed over to this plug-in check function. This has to be defined in the Runtime Input view of the Check Routine explorer.

The correct order is:

Input 1: Original image.
Input 2: ROI that was used as input for Unroll ROI.
Input 3: Unrolled image.
Input 4: ROIs that have to be reverse transformed.

This order will not be correct by default in many cases and has to be checked and defined manually by the user.

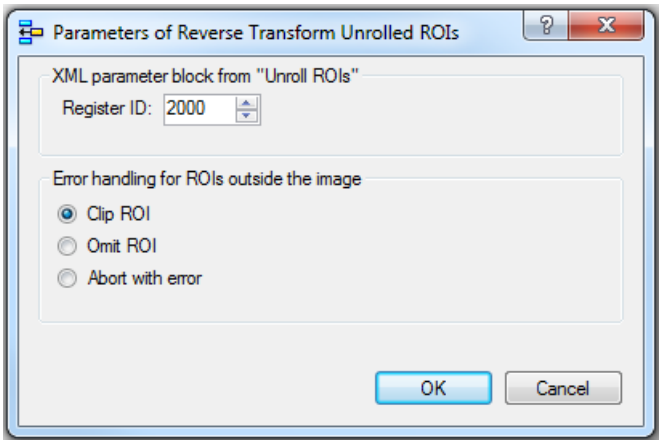
The following illustration shows the order of the input data in the initial state (left side) and the manually corrected order (right side) of the input data. The input data that had to be corrected was the ROI that was used as Input for the Check Function "Unroll ROIs".


Check Routine Explorer
Structure
Runtime Input
Flow
Transfer Image
Define ROIs
Unroll ROI
Define ROIs
Determine Threshold
Create ROIs by Thresholding
Reverse transform Unrolled ROIs
Input data
Output data
Check Routine Explorer
Structure
Runtime Input
Flow
Transfer Image
Define ROIs
Unroll ROI
Define ROIs
Determine Threshold
Create ROIs by Thresholding
Reverse transform Unrolled ROIs
Input data
Output data

Reverse Transform Unrolled ROIs: Parameter Dialog

This plug-in check function has a **Parameter** dialog.

Screenshot of Parameter Dialog



The **Parameter** dialog contains the following elements:

Element	Description
XML parameter block from "Unroll ROIs"	Set the data source of the input parameters.
Register ID	Register ID that contains the XML parameter block of "Unroll ROIs". Has to be of data type String.
Error handling for ROIs outside the image	Defines how reverse transformed ROIs that are outside the original images should be treated.
Clip ROI	Reverse transformed ROI will be clipped by original image borders, if it is outside the original image.
Omit ROI	Reverse transformed ROI will be omitted if it is outside the original image and has no contribution to the output of the check function.
Abort with error	Check function aborts with error, if at least one reverse transformed unrolled ROI is outside the original image.

Reverse Transform Unrolled ROIs: Visualizations

This section describes the result visualizations the check function "Reverse Transform Unrolled ROIs" provides.

Element	Description
Transformed ROIs	Displays the original image and a list of reverse transformed ROIs, that accord to the output data of the check function.
Original ROIs	Displays the original image and the original ROI that was used as input for the Unroll ROIs check function.
Unrolled ROIs	Displays the unrolled image and a list of unrolled ROIs.

Save ROIs: Introduction

Function overview

This Plug-In function saves ROIs as an XML document. It can be used to preserve a list of ROIs, e.g. teached objects, between NeuroCheck sessions. The saved XML document can be loaded again by the check function Load ROIs. As a saving destination the user may specify:

- File with a fixed path;
- File with a path taken dynamically from the data register;
- Data register of type String to save the document as a singe XML string.

In time-critical applications you can turn on the asynchronous file saving option to minimize the total execution time. In this case the saving operation will continue in parallel with the check routine execution.

Note that only one ROI file saving operation can run at a time. Consequent asynchronous file saving requests will run sequentially.

Input data

This check function requires an image and a list of ROIs as input data objects.

Output data

None. (An XML document is saved either to the file or to the data register)

Result view

The result view shows the input image and ROIs.

Properties

 Check function group Plug-In

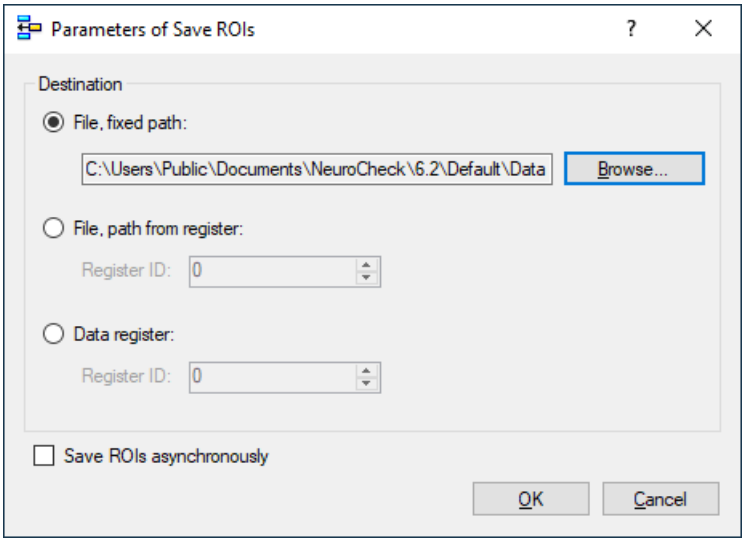
 The check function has a Parameter Dialog.

Save ROIs: Parameter Dialog

Parameter

This plug-in check function has a **Parameter** dialog.

Screenshot of Parameter Dialog



The **Parameter** dialog contains the following elements to specify the saving options:

Element	Description
File, fixed path	Select an existing file path or create a new one.
File, path from register	Select a register cell containing a file path as String.
Data register	Specify the target data register cell of type String
Save ROIs asynchronously	Check if the file saving operation should run asynchronously.

Tile ROIs: Introduction

Function

This check function can be used to split ROIs into smaller tiles.

Each of the input ROIs is cut into tiles of the size specified by the parameters. Because the original ROI can rarely be covered evenly with fixed sized tiles, the check function has two main modes:

- Stay inside ROI borders
- Breach ROI borders

In the first mode the resulting tile ROIs will stay inside the original ROI borders by leaving a gap between the last tile and the original border. The second mode will add one tile more, to make sure the original ROI is completely covered by tiles. If the resulting ROIs would leave the image borders the check function will abort with an error.

The check function allows to specify an overlap between to adjacent tiles. This overlap may also be negative to provide a gap.

Input data

This check function requires a list of ROIs as input data object.

Output data

The list of ROIs, created by tiling the input ROIs.

Result view

The result view shows the new output ROIs.

Properties

 Check function group Plug-In

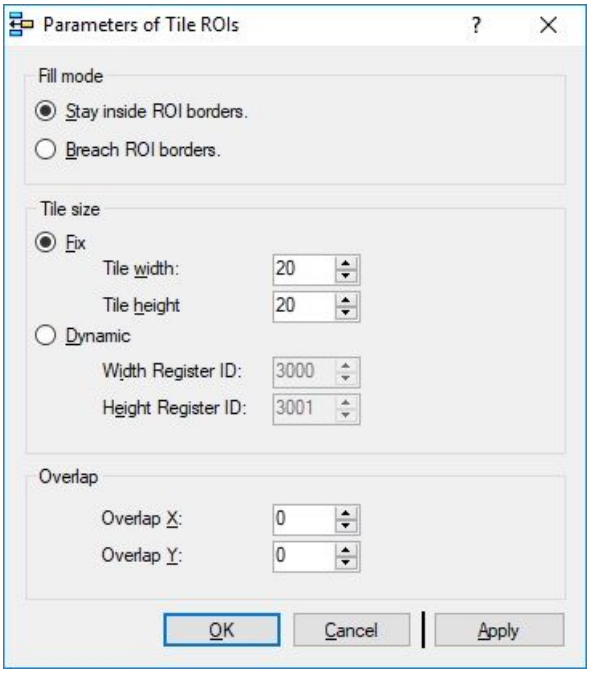
 The check function has a Parameter Dialog.

 The check function has own result Visualizations.

Tile ROIs: Parameter Dialog

This plug-in check function has a **Parameter** dialog.

Screenshot of Parameter Dialog



The **Parameter** dialog contains the following elements:

Element	Description
Stay inside ROI borders	The resulting ROIs will stay inside the original ROIs borders when choosing this mode.
Breach ROI borders	The resulting ROIs will cover the original ROI completely by overlapping its borders.
Fix	The tile size and width can be specified directly in the parameter dialog.
Tile width	The width of the resulting tile ROIs in pixels.
Tile height	The height of the resulting tile ROIs in pixels.
Dynamic	The tile size and width will be dynamically loaded from register.
Width Register ID	ID of the Register that holds the tile width.
Height Register ID	ID of the Register that holds the tile height.
OverlapX	Horizontal overlap between two tile ROIs. May be negative to create a gap.
OverlapY	Vertical overlap between two tile ROIs. May be negative to create a gap.

Tile ROIs: Visualizations

This section describes the result visualizations the check function "Tile ROIs" provides.

Element	Description
Output ROIs	The resulting tile ROIs.
Input ROIs	The input ROIs.

Support Services

For technical support, please contact your local NeuroCheck partner or NeuroCheck GmbH:

Phone: +49 (0) 7146 - 89 56-40

E-Mail: support@neurocheck.com

Web: www.neurocheck.com

Before contacting us, please provide some important information about your system:

Information about your NeuroCheck installation and your PC setup:

- Use the NeuroCheck Diagnostics tool to check your installation and computer configuration.
- The NeuroCheck Diagnostics is installed in the "Tools" folder within your NeuroCheck installation.

Log file information:

- Logging for NeuroCheck can be activated in **System** > **Software Settings** > **Diagnosis** > **Logging**.