

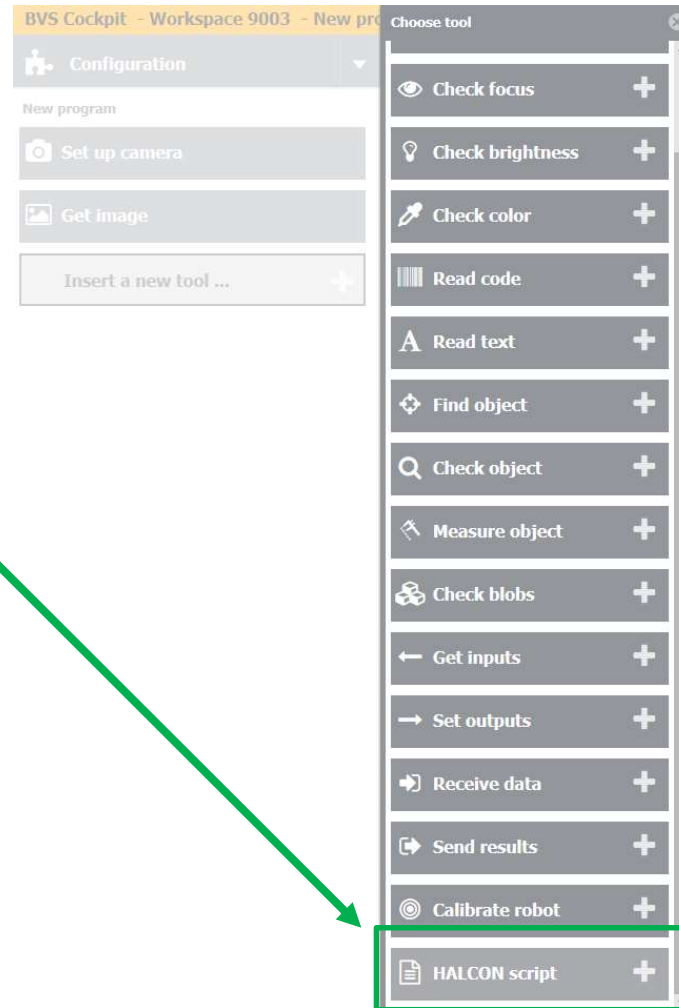
**BALLUFF**

# Step-by-Step Guide

**Measurements**

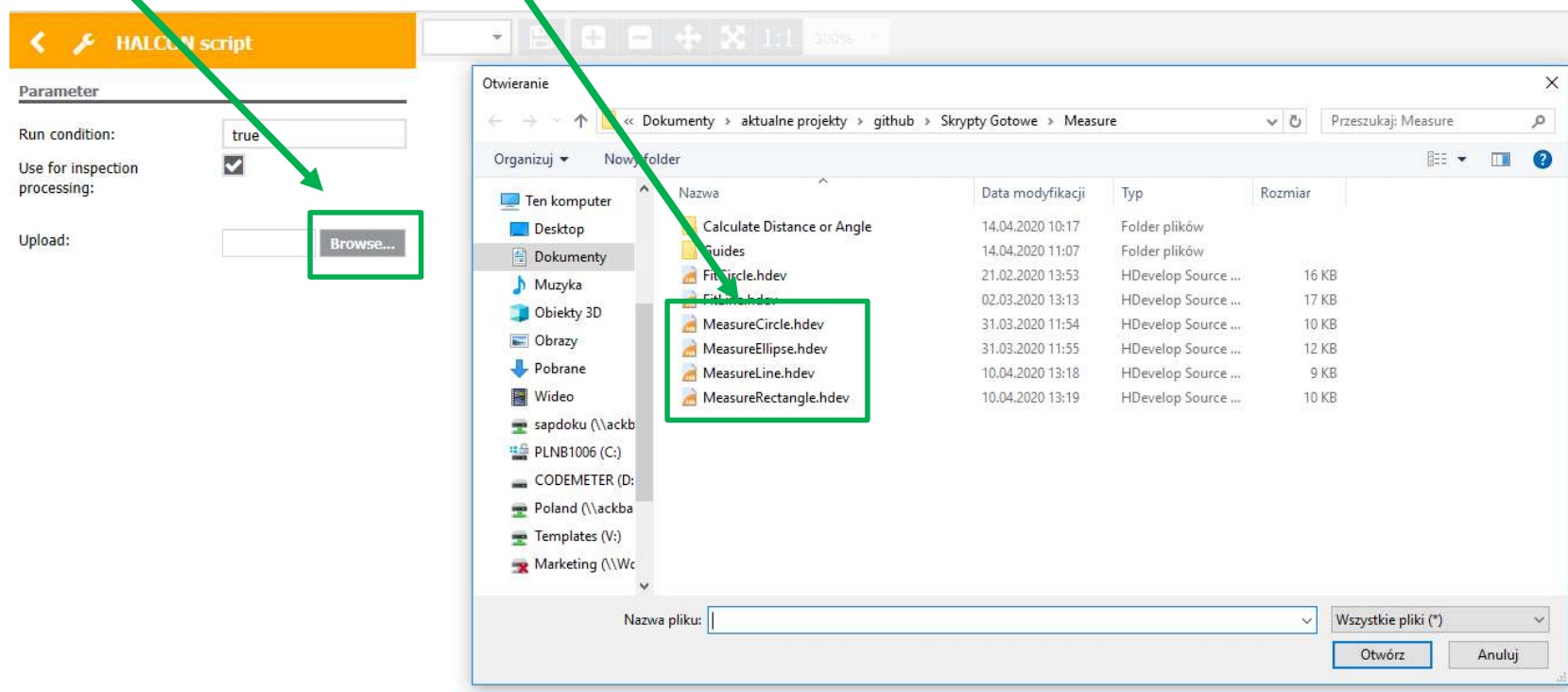
## Step 1

Add new tool – Halcon Script – in your inspection program.



## Step 2

Click „Browse” and choose „.hdev” file depending on what do you want to measure.



## Step 3

If you want to measure circle – set the following parameters:

- a) CenterX – X coordinate of circle's center point
- b) CenterY – Y coordinate of circle's center point
- c) Radius – circle radius
- d) ArcAngleStart – arc's starting angle
- e) ArcAngleEnd – arc's ending angle

It is possible to set outputs from previous tools (for example Check blobs) as input values.

**Measure Circle**

Parameter	
Run condition:	<input type="text" value="true"/>
Use for inspection processing:	<input checked="" type="checkbox"/>
Upload:	<input type="button" value="Measure"/> <input type="button" value="Browse..."/>
ArrayCount:	<input type="text" value="1"/> <input type="button" value="-"/> <input type="button" value="+"/>
InputImage:	<input type="text" value="Get image.Output"/>
CenterX:	<input type="text" value="Check_blobs.Position_"/>
CenterY:	<input type="text" value="Check_blobs.Position_"/>
Radius:	<input type="text" value="(Check_blobs.[Featur"/>
ArcAngleStart:	<input type="text" value="0,000"/>
ArcAngleEnd:	<input type="text" value="360"/>

## Step 3

If you want to measure ellipse – set the following parameters:

- a) CenterX – X coordinate of ellipse's center point
- b) CenterY – Y coordinate of ellipse's center point
- c) Phi – orientation of the main axis
- d) Radius1 – length of the larger half axis
- e) Radius2 – length of the smaller half axis
- f) ArcAngleStart – elliptic arc's starting angle
- g) ArcAngleEnd – elliptic arc's ending angle

It is possible to set outputs from previous tools (for example Check blobs) as input values.

**Measure Ellipse**

---

**Parameter**

Run condition:	<input type="text" value="true"/>
Use for inspection processing:	<input checked="" type="checkbox"/>
Upload:	<input type="button" value="Measure"/> <input type="button" value="Browse..."/>
ArrayCount:	<input type="text" value="1"/> <input type="button" value="-"/> <input type="button" value="+"/>
InputImage:	<input type="text" value="Get image.Output"/>

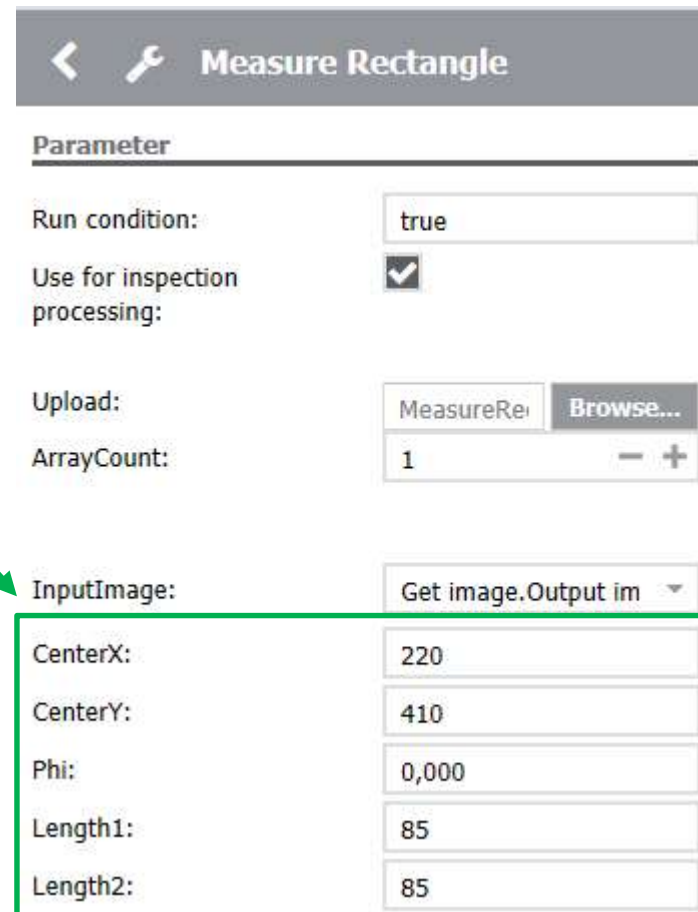
CenterX:	<input type="text" value="Check_blobs.Position_"/>
CenterY:	<input type="text" value="Check_blobs.Position_"/>
Phi:	<input type="text" value="0,000"/>
Radius1:	<input type="text" value="Check_blobs.[Feature"/>
Radius2:	<input type="text" value="Check_blobs.[Feature"/>
AngleStart:	<input type="text" value="0,000"/>
AngleEnd:	<input type="text" value="360"/>

## Step 3

If you want to measure rectangle – set the following parameters:

- a) CenterX – X coordinate of rectangle's center point
- b) CenterY – Y coordinate of rectangle's center point
- c) Phi – orientation of the main axis
- d) Length1 – length of the larger half edge
- e) Length2 – length of the smaller half edge

It is possible to set outputs from previous tools (for example Check blobs) as input values.



**Measure Rectangle**

**Parameter**

Run condition:

Use for inspection processing: ☒

Upload:  **Browse...**

ArrayCount:  **- +**

InputImage:

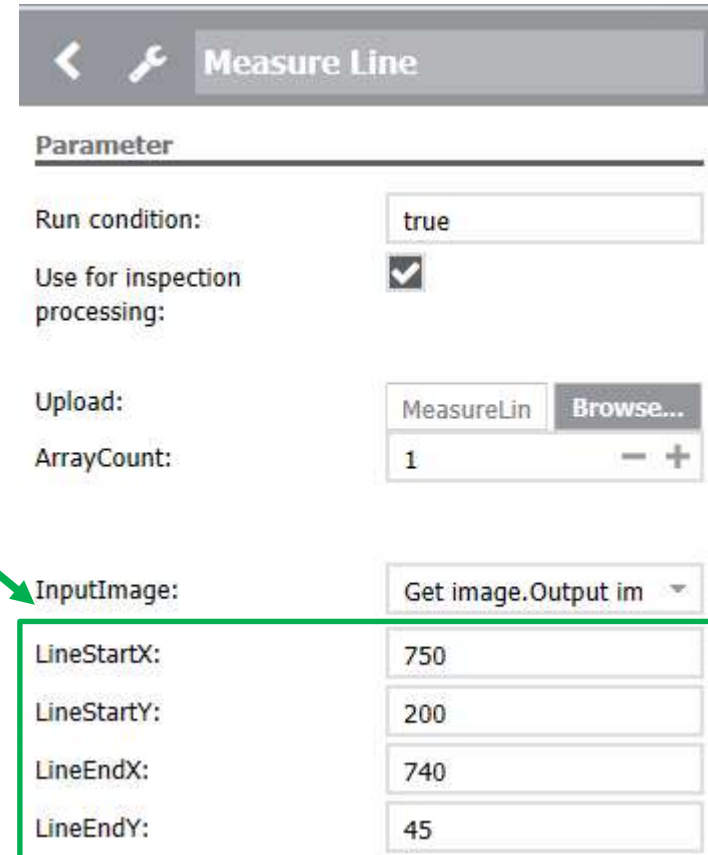
CenterX:	<input type="text" value="220"/>
CenterY:	<input type="text" value="410"/>
Phi:	<input type="text" value="0,000"/>
Length1:	<input type="text" value="85"/>
Length2:	<input type="text" value="85"/>

## Step 3

If you want to measure line – set the following parameters:

- a) LineStartX – X coordinate of line's start point
- b) LineStartY – Y coordinate of line's start point
- c) LineEndX – X coordinate of line's end point
- d) LineEndY – Y coordinate of line's end point

It is possible to set outputs from previous tools (for example Check blobs) as input values.



**Measure Line**

**Parameter**

Run condition:	true
Use for inspection processing:	<input checked="" type="checkbox"/>
Upload:	MeasureLin <b>Browse...</b>
ArrayCount:	1 - +
InputImage:	Get image.Output im ▾
LineStartX:	750
LineStartY:	200
LineEndX:	740
LineEndY:	45

## Step 4

Adjust edge parameters:

- a) MeasureLength1 and MeasureLength2 – lengths of the measuring rectangle (visible as MeasureRegions variable on the image)
- b) MeasureSigma – smoothing
- c) MeasureThreshold – minimum edge contrast
- d) PointOrder – counterclockwise (positive) direction or clockwise (negative) direction of the circular arc

MeasureLength1:	<input type="text" value="20,000"/>
MeasureLength2:	<input type="text" value="5,000"/>
MeasureSigma:	<input type="text" value="1,000"/>
MeasureThreshold:	<input type="text" value="30,000"/>
PointOrder:	<input type="text" value="positive"/> ▼



## Step 5

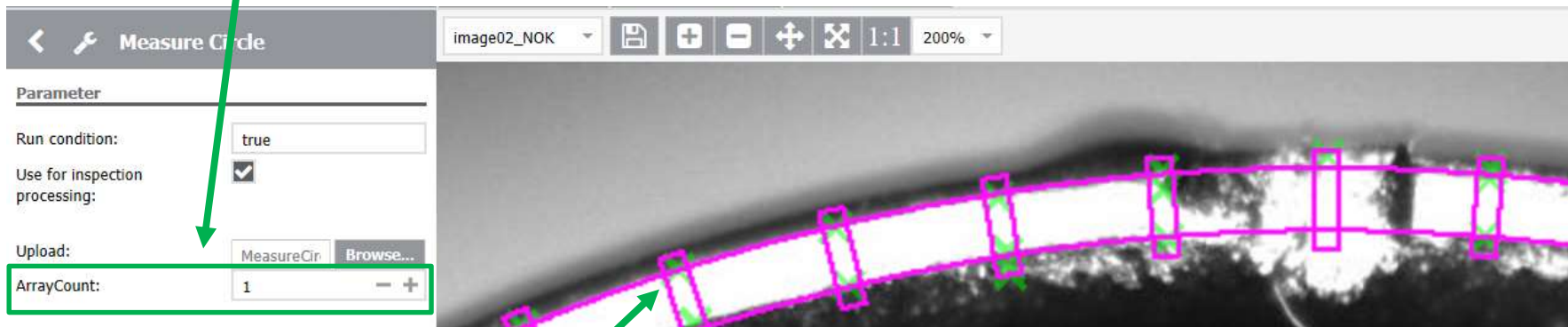
Adjust metrology parameters:

- a) NumberMeasures – number of measuring rectangles
- b) MeasureSelect:
  - First – only first edge within measuring rectangle will be found
  - Last – only last edge within measuring rectangle will be found
  - All – all of the edges within measuring rectangle will be found
- c) MeasureTransition:
  - Positive – only edges with polarity from dark to light will be found
  - Negative – only edges with polarity from light to dark will be found
  - All – edges with both polarity will be found
- d) MinScore – minimum score for a measure instance to be considered as valid
- e) NumberInstances – number of measuring instances
- f) DistanceThreshold – single found edge is considered to be a part of the fitted circle if it doesn't exceed this parameter

NumberMeasures:	<input type="text" value="3"/> <span>–</span> <span>+</span>
MeasureSelect:	<input type="text" value="all"/> ▼
MeasureTransition:	<input type="text" value="all"/> ▼
MinScore:	<input type="text" value="0,7"/> <span>–</span> <span>+</span>
NumberInstances:	<input type="text" value="1"/> <span>–</span> <span>+</span>
DistanceThreshold:	<input type="text" value="3,5"/> <span>–</span> <span>+</span>

## Step 6

Set the number of instances that you want to get as an output.



Found edges will be displayed as crosses (MeasurePoints output variable) on the image.

## Step 7

Fited circle will be displayed in Circle output variable (as circle on the image).

Available output parameters:

- MeasuredCenterX – X coordinate of the circle's center point
- MeasuredCenterY – Y coordinate of the circle's center point
- MeasuredRadius – circle radius
- MeasuredCenterXCalibrated – X coordinate of the circle's center point after calibration
- MeasuredCenterYCalibrated – Y coordinate of the circle's center point after calibration
- MeasuredRadiusCalibrated – circle radius after calibration

**Parameter**

Run condition: ☐ true

Use for inspection processing: ☒

Upload:

ArrayCount:

InputImage:

CenterX:

CenterY:

Radius:

ArcAngleStart:

ArcAngleEnd:

MeasureLength1:

MeasureLength2:

MeasureSigma:

MeasureThreshold:

PointOrder:

NumberMeasures:

MeasureSelect:

MeasureTransition:

MinScore:

NumberInstances:

DistanceThreshold:

UseCalibration: ☒

**Image Resolution: 1968 x 1912**

**Results - Measure Circle**

Name	Value
Tool processing	Successful
Result message	
Execution duration [ms]	18,451
MeasurePoints	
Circle	
MeasureRegions	
MeasuredCenterX 1	988,082
MeasuredCenterY 1	967,338
MeasuredRadius 1	852,346
MeasuredCenterXCalibrated 1	0
MeasuredCenterYCalibrated 1	0
MeasuredRadiusCalibrated 1	0

## Step 7

Fitted ellipse will be displayed in Ellipse output variable (as ellipse on the image).

Available output parameters:

- MeasuredCenterX – X coordinate of the ellipse's center point
- MeasuredCenterY – Y coordinate of the ellipse's center point
- MeasuredPhi – measured orientation of the main axis
- MeasuredRadius1 – measured length of the larger half axis
- MeasuredRadius2 – measured length of the smaller half axis
- MeasuredCenterX – X coordinate of the ellipse's center point after calibration
- MeasuredCenterY – Y coordinate of the ellipse's center point after calibration
- MeasuredPhi – measured orientation of the main axis after calibration
- MeasuredRadius1 – measured length of the larger half axis after calibration
- MeasuredRadius2 – measured length of the smaller half axis after calibration

**Measure Ellipse**

Parameter

Run condition: ☐ true

Use for inspection processing: ☒

Upload:

ArrayCount: 1

InputImage:

CenterX:

CenterY:

Phi:

Radius1:

Radius2:

AngleStart:

AngleEnd:

MeasureLength1:

MeasureLength2:

MeasureSigma:

MeasureThreshold:

PointOrder:

MeasureDistance:

NumberMeasures:

MeasureSelect:

MeasureTransition:

MinScore:

NumberInstances:

Image Resolution: 1968 x 1912

**Results - Measure Ellipse**

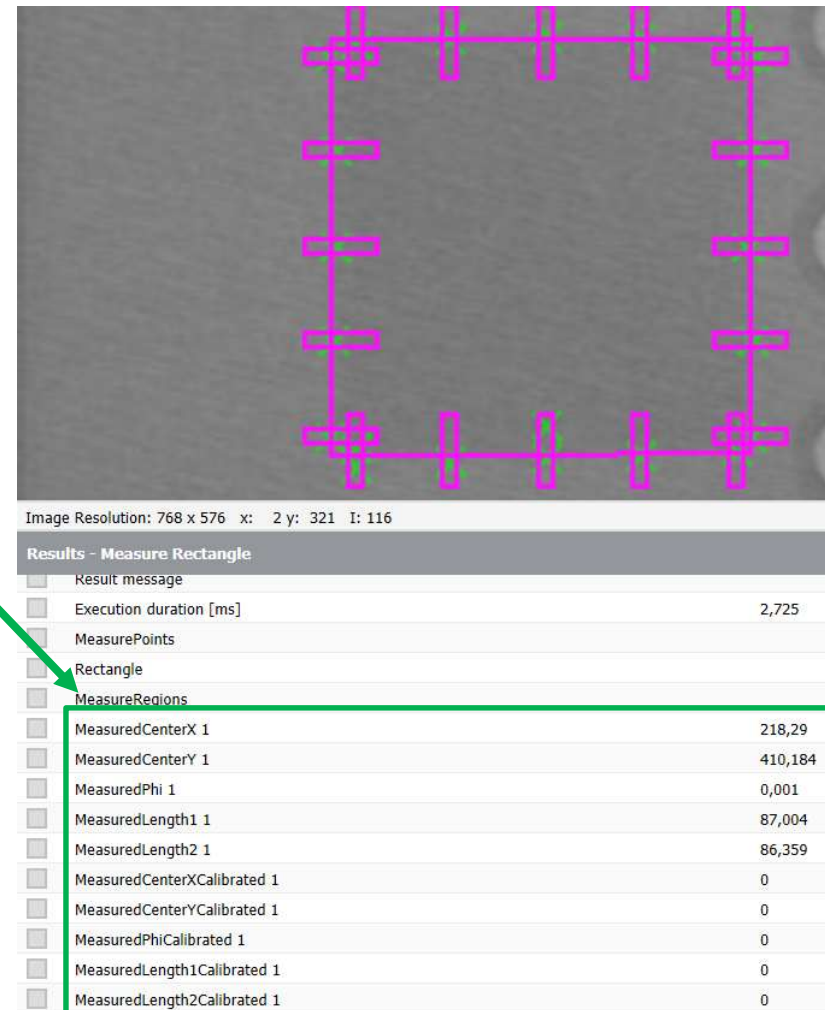
Name	Value
Tool processing	Successful
Result message	
Execution duration [ms]	101,517
MeasurePoints	
Ellipse	
MeasureRegions	
MeasuredCenterX 1	989,049
MeasuredCenterY 1	968,208
MeasuredPhi 1	-0,751
MeasuredRadius1 1	829,803
MeasuredRadius2 1	826,445
MeasuredCenterXCalibrated 1	0
MeasuredCenterYCalibrated 1	0
MeasuredPhiCalibrated 1	0
MeasuredRadius1Calibrated 1	0
MeasuredRadius2Calibrated 1	0

## Step 7

Fitted rectangle will be displayed in Rectangle output variable (as rectangle on the image).

Available output parameters:

- MeasuredCenterX – measured X coordinate of rectangle's center point
- MeasuredCenterY – measured Y coordinate of rectangle's center point
- MeasuredPhi – measured orientation of the main axis
- MeasuredLength1 – measured length of the larger half edge
- MeasuredLength2 – measured length of the smaller half edge
- MeasuredCenterXCalibrated – measured X coordinate of rectangle's center point after calibration
- MeasuredCenterYCalibrated – measured Y coordinate of rectangle's center point after calibration
- MeasuredPhiCalibrated – measured orientation of the main axis after calibration
- MeasuredLength1Calibrated – measured length of the larger half Edge after calibration
- MeasuredLength2Calibrated – measured length of the smaller half Edge after calibration



## Step 7

Fitted line will be displayed in Line output variable (as line on the image).

Available output parameters:

- a) MeasuredLineStartX – measured X coordinate of line's start point
- b) MeasuredLineStartY – measured Y coordinate of line's start point
- c) MeasuredLineEndX – measured X coordinate of line's end point
- d) MeasuredLineEndY – measured Y coordinate of line's end point
- e) MeasuredLineStartXCalibrated – measured X coordinate of line's start point after calibration
- f) MeasuredLineStartYCalibrated – measured Y coordinate of line's start point after calibration
- g) MeasuredLineEndXCalibrated – measured X coordinate of line's end point after calibration
- h) MeasuredLineEndYCalibrated – measured Y coordinate of line's end point after calibration



Image Resolution: 1600 x 1200

Results - Measure Line	
<input type="checkbox"/>	Result message
<input type="checkbox"/>	Execution duration [ms] 0,986
<input type="checkbox"/>	MeasurePoints
<input type="checkbox"/>	Line
<input type="checkbox"/>	MeasureRegions
<input type="checkbox"/>	MeasuredLineStartX 1 750,384
<input type="checkbox"/>	MeasuredLineStartY 1 194,913
<input type="checkbox"/>	MeasuredLineEndX 1 750,542
<input type="checkbox"/>	MeasuredLineEndY 1 48,322
<input type="checkbox"/>	MeasuredLineStartXCalibrated 1 0
<input type="checkbox"/>	MeasuredLineStartYCalibrated 1 0
<input type="checkbox"/>	MeasuredLineEndXCalibrated 1 0
<input type="checkbox"/>	MeasuredLineEndYCalibrated 1 0

## Step 8

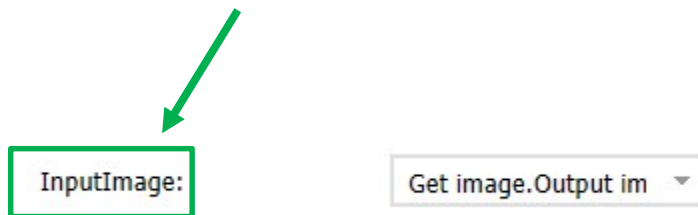
Calibrated output parameters have proper values if the UseCalibration variable is checked and calibration data is available.

UseCalibration:



## Step 9

It is also possible to use locator from previous tools – in InputImage choose output image from locator tool (for example Find object tool).





**BALLUFF**

# BALLUFF A GLOBAL PROMISE

 *innovating automation*