LIEBHERR

LHRobotics.Vision Quick Start Guide

Version

Version Nr.	Changes	Date
1.0	First version of the Quick Start Guide	30.08.2022
1.1	Version 3.4 Description of AI feature (chapter 6.2.2) Described in more detail: Hand-eye-calibration, simulation, troubleshooting	03.02.2023
1.2	Version 3.5 Adjustable simulation camera	23.05.2023

Contents

1. Installation LHRobotics.Vision

1.1 System requirements

To run the LHRobotics.Vision software, a Windows 10 or Windows 11 computer with the following configuration is required:

Minimum

OS:	Windows 10/11 64 Bit
Processor:	Minimum 4 Cores (e.g. Intel Core i5)
Memory:	32 GB RAM
Graphics card:	Intern
Storage:	500 GB

Recommended

OS:	Windows 10/11 64 Bit
Processor:	Minimum 8 Cores (e.g. Intel Core i7 newest generation)
Memory:	64 GB RAM
Graphics card:	E.g. GeForce GTX 1080
Storage:	500 GB SSD

1.2 Requirements regarding the robot

To ensure the functionality of the system, the robot must have certain features. These are as follows:

Robot	Option
KUKA	• Measured with absolute accuracy • Multi Submit for the use of several submit programs • KUKA Ethernet KRL for TCP/IP data communication • OrangeApps TrafoAdvanced (can be obtained from Liebherr)
ABB	• Absolute accuracy, so that TCP is accurate to +/- 1mm • PC interface for communication between robot control and PC • Multitasking - so that other processes can be controlled simultaneously with the robot movement
Fanuc	• Ascii Program Loader • Absolutely accurate measurement • KAREL (included as standard in EU; must be ordered separately in USA) • User Socket Messaging for TCP/IP data communication • Singularity Avoidance, allowing robot to travel near a singularity without error message

1.3 Installing the dependencies

First of all please unpack the enclosed zip file. All required components are included there.
The following dependencies are required for the software to start:

- **HASPUserSetup.exe**
  Required for successful detection of the license dongle (not required for internal demo systems)
- **vc_redistx64.exe**
  Required for running the LHRobotics vision software
- **Sensor SDK e.g. ZividSetup_2.x.x+xxxxxxxx-x.exe**
  Required for communication and sensor setup

These files are located in the subfolder ‘**Dependencies**’.
Install all the dependencies one by one.

1.4 Installation of the software

Unzip the enclosed zip file (**Release64_x_x_x.zip**) to a location of your choice, e.g. on the desktop.

- This will be the archive for your software and all projects
- **Attention**: The location should not be changed afterwards!

The folder structure should now look like this:

- documentation
- lang
- modules
- platforms
- qt
- robots
- git2.dll
- hasp_windows_x64_35899.dll
- LHRobotics
- LHRoboticsCore
- libgcc_s_seh-1.dll
- libnlopt.dll
- libstdc++-6.dll
- libwinpthread-1.dll
- Open3D.dll
- opencv_world420.dll
- qt.conf
- Qt5Core.dll
- Qt5Gui.dll
- Qt5Widgets.dll
- QtWebEngineProcess

Note that the 'projects' folder is created only after the first start of the software.

- All your projects are stored there
- This folder can also be used for data backup.

1.5 Installing the simulation environment (optional)

The following dependencies are required for the simulation to run.

- **EnsensoSDK-3.x.xxx.exe**
  Needed to create a realistic point cloud.
- **CoppeliaSim_Player_V4_x_x_Setup.exe**
  Serves as a simulation environment with integrated physics engine for chaotic arrangement of parts.

Both files are located in the subfolder 'Dependencies.
Please install both dependencies one after another.

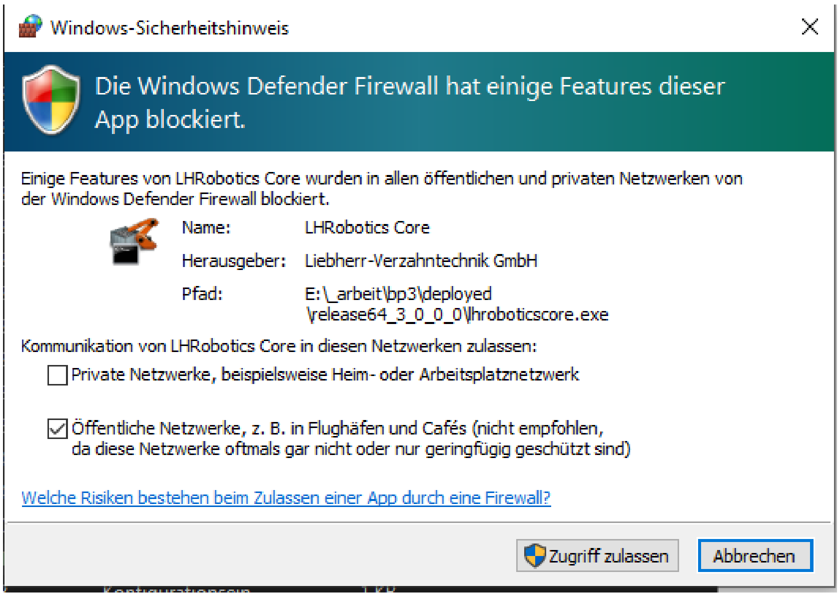
1.6 Initial setup

Plug the enclosed hardware dongle into a free USB port, otherwise the software will not start.

The software is started via the following file in your installation folder:

LHRobotics.exe

When opening the software for the first time, the following dialog should open:



If the software is to be used as a productive system, access must be allowed, otherwise the robot will not be able to connect to the software later.

2 separate programs are started:

- **LHRobotics GUI:**
  Used to configure the program, interface for setup.
- **LHRobotics Core:**
  Used for robot communication
  For productive systems, the GUI is not necessarily needed.

Access documentation:
On each page there is an icon in the upper right corner to call up the online help.

At the first start a new project is created automatically. This is saved in the 'projects' folder. The folder can be saved to another location.
In this case, the storage path 'ProjectPath=…' in the LHRoboticsCore.ini must be adjusted.

1.7 Potential problems

1.7.1 Problems when starting the software

If one of the following two messages appears when starting the software, please install all dependencies (see 1.2).

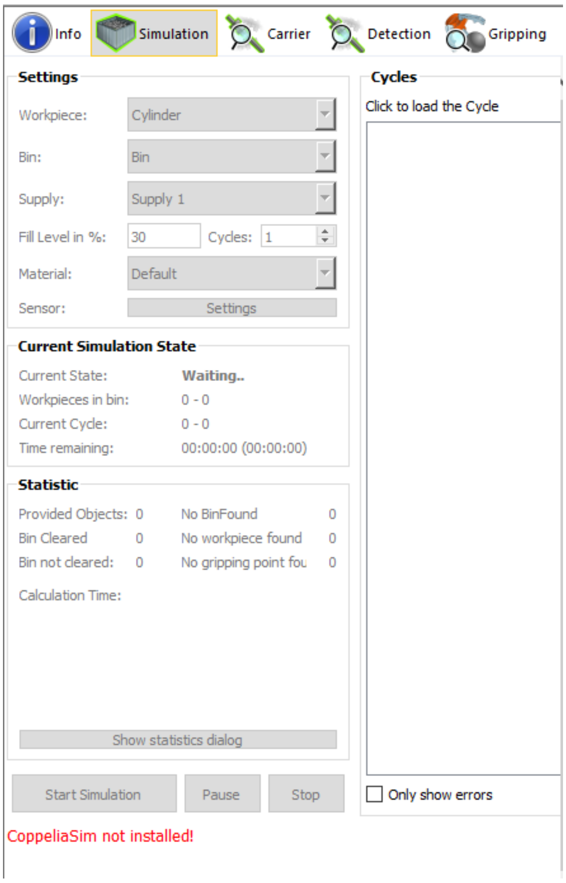
- mfc140u.dll is missing
- MSCVP140.dll is missing

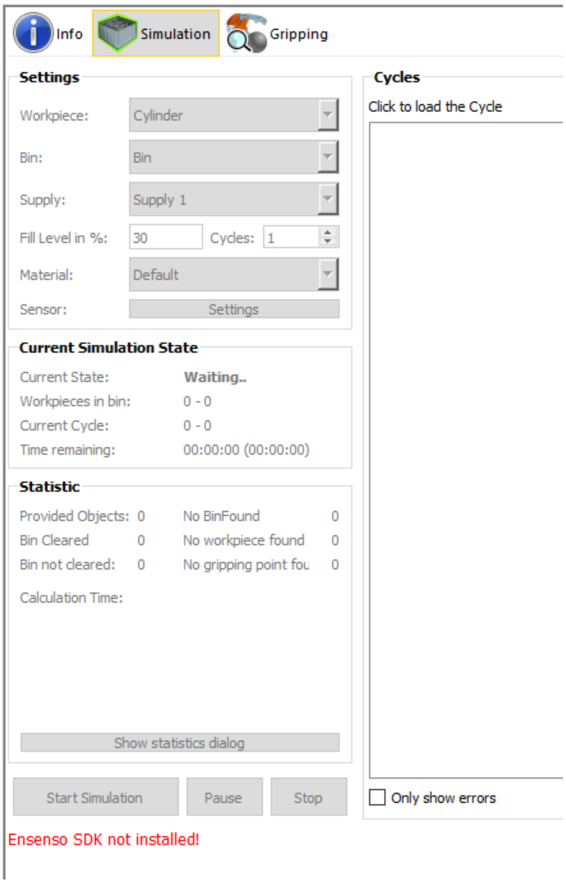
If the following message appears, please install all dependencies (see 1.2) and make sure that the enclosed hardware dongle is connected to the computer.

```
Licence error: USB Dongle missing
Dongle info: No dongle information available!
c2v file created. (C:\Users\LVT\Desktop\__Installation\Release64_3_0_1_0\Release64_3_0_1_0\bp3_dongle_info.c2v)
Drücken Sie eine beliebige Taste . . .
```

1.7.2 Problems when starting the offline simulation

If one of the following messages appear, please install all dependencies (see 1.2) for offline simulation and restart the program.





2. Installation of the sensor

The sensor must be installed mechanically and electrically according to the operating instructions of the camera supplier. An SDK (Software Development Kit) must be installed to operate the sensor and to change settings for image captures.

The operating instructions for these steps can be found in a separate document. Select the appropriate instruction for your camera and its SDK.

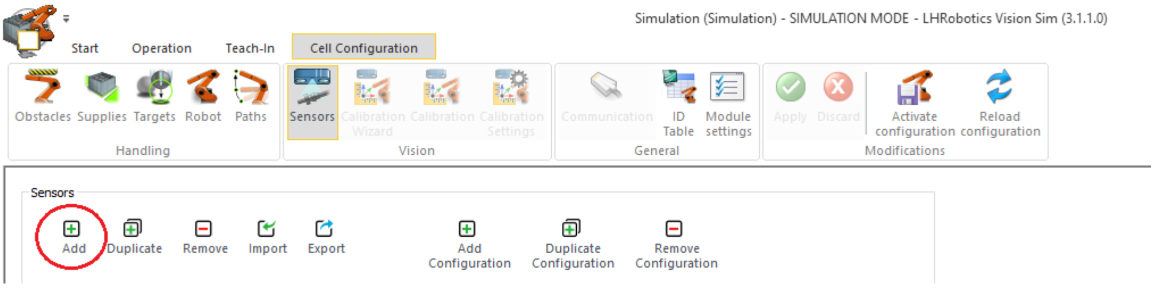
Note on the calibration of Zivid cameras
The calibration 'Infield correction' is described in the Zivid Knowledge Base. To start the correction, the ZividExperimentalInfieldCorrection.exe file must be opened from the Windows console in the Zivid SDK installation path.

- Start the Windows console
  - Enter 'CMD' in the program search and start cmd.exe
  - Or: press Windows key and 'R' simultaneously
- Navigate to the Zivid installation directory by typing and confirming with Enter key:
  `cd C:\Program Files\Zivid\bin\`
- Start the exe file by entering of … (confirm with Enter key)
  `\ZividExperimentalInfieldCorrection.exe`

3. Creating the sensor in the software

3.1 Adding the Zivid Sensors

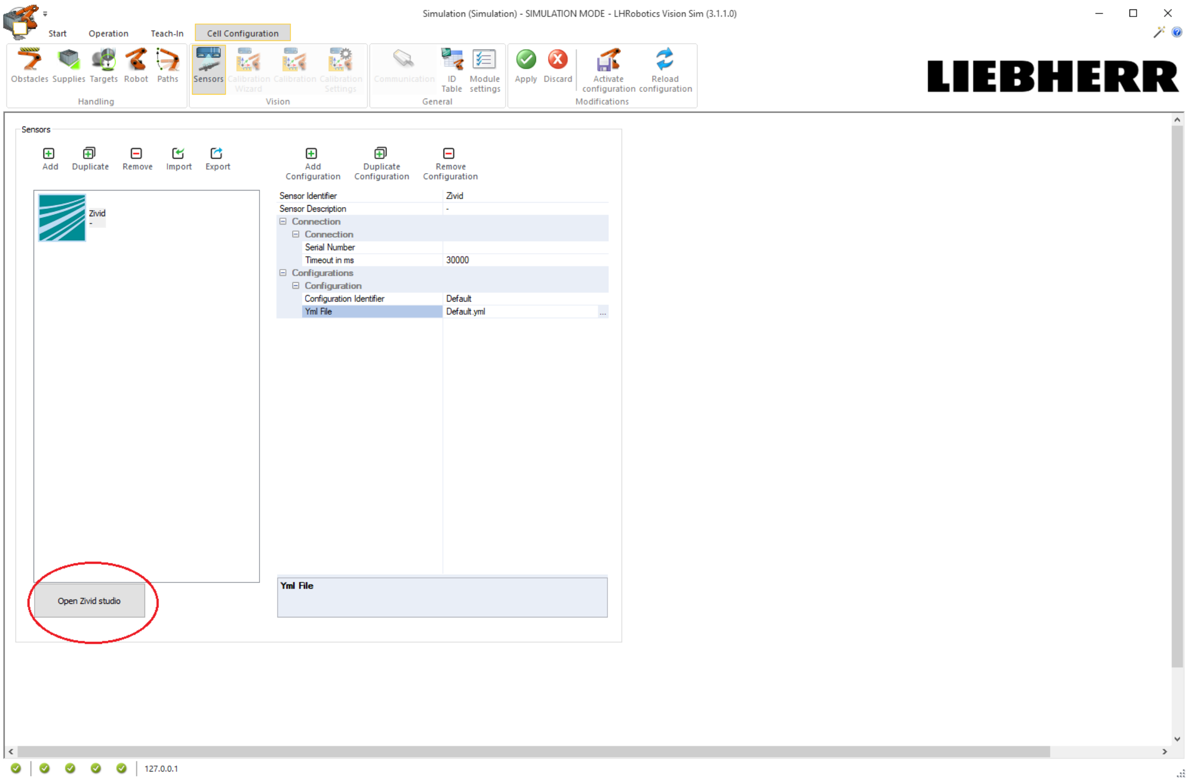
Go to **Cell Configuration** > Sensors.
If no sensor has been added yet, click on the **Add** button and select the corresponding sensor.



3.2 Parameterization of the Zivid sensor

To start the SDK you have two options:

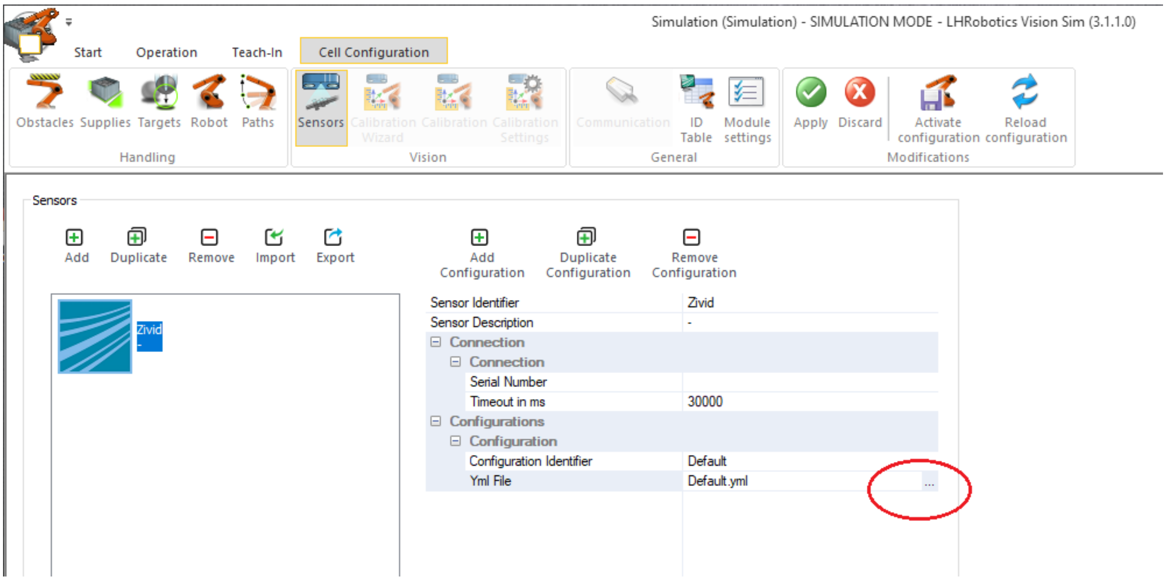
1. Via the Windows start menu
2. Directly via LHRobotics.Vision. For this option the respective sensor must be selected in the left-hand list.



For parameterization of the sensor, we refer to the user manual of the sensor manufacturer see chapter 2.

3.3 Adopting the parameters (Sensor SDK → LHRobotics.Vision)

1. Sensor SDK
   - Exporting the parameters via the menu item **File / Export Capture Settings**
2. LHRobotics.Vision
   - Take over the exported parameter file (*.yml)



4. Communication with the robot

4.1 Interface

The communication between robot and LHRobotics.Vision takes place in ASCII format via the TCP/IP interface.

Port: 1234
IP-Adresse: That of the computer

LHRobotics.Vision serves as server and the robot as client.

If no connection between LHRobotics.Vision and the robot is established, please check your firewall settings!
If the robot is not addressable from the computer, try to change the following setting.

1. The robot interface must be set up as follows:
   - Communication protocol for the exchange of commands / information via TCP/IP.
   - Liebherr provides an interface description. Here, the messages (approx. 20-25 pieces) which are exchanged between robot and software are described as in the following example:

Messages to LHRobotics.Vision

Message 001 - Request Object

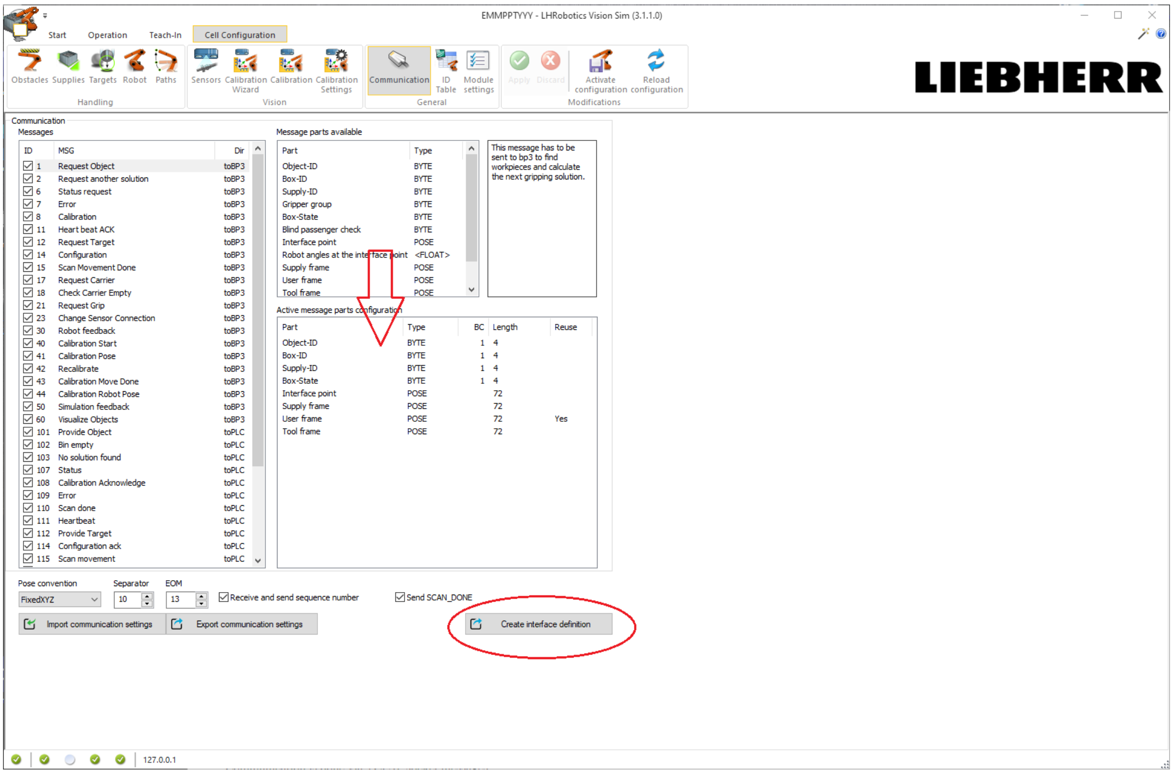
This message has to be sent to bp3 to find workpieces and calculate the next gripping solution.

	Name	Type	Length	Description / value
Header:	Message header	MSGSTR	4	**msg**
	Message identifier	BYTE	4	**001**
Body:	Object-ID	BYTE	4	The ID of the object to search for. (1='Planetary_carrier_8', 2='Gear_wheel_8_Axis', 3='Tracklink_8_Axis', 4='Gear_Ring_6_Axis', 5='3_Cylinder_Crankshaft_6_Axis', 6='4_Cylinder_Crankshaft_8_Axis', 7='3_Cylinder_Crankshaft_8_Axis', 8='Gear_Ring_8_Axis', 9='Trinkflasche')
	Box-ID	BYTE	4	The box type to pick from. (1='Bin')
	Supply-ID	BYTE	4	The current supply position.
	Box-State	BYTE	4	1 = no changes; 2 = box changed; 3 = door was opened.
	Interface point	POSE	72	Last pose defined by robot. This is where bp3 starts its path planing.
	Supply frame	POSE	72	Supply frame in world where the bin is searched
	Tool frame	POSE	72	The pose defines the tool frame with the respect to the robot's flange
	Example:	msg□001□002□002□002□002□-1234.56789□-1234.56789□-1234.56789□-1234.56789□-1234.56789□-1234.56789□-1234.56789□-1234.56789□-1234.56789□-1234.56789□-1234.56789□-1234.56789□-1234.56789□-1234.56789□-1234.56789□-1234.56789□-1234.56789□-1234.56789□■		

2. The content of these messages can be customized by the user.

To view or change the individual messages i.e. requests, open the dialog under **Cell Configuration / Communication.**

Attention: Only change something here if it is necessary.
If changes are made, the robot program must also be adapted.



Here, each request can be customized.
The parts of the individual messages can be assembled individually by drag & drop

An interface definition can also be exported as an HTML page for easy implementation in the robot program.

4.2 Robot program

LHRobotics.Vision usually works as a subsystem for a robot. LHRobotics.Vision processes request messages from the robot, performs calculations and returns a response message. A typical sequence is as follows:

1. Robot sends REQUEST_OBJECT message to get a gripping path for the next workpiece. The message contains the parameters set in LHRobotics.Vision, including ...
   - the supply / area where workpieces are searched
   - the workpiece to be searched
   - the workpiece carrier / bin
   - whether a new bin has been provided (only necessary for multiple picks without rescanning)

Robot Message	1 - Request object
Supply	1 - Supply 1
Workpiece	3 - Blank gear
Bin	1 - Bin
Box State	1 - no changes

2. LHRobotics.Vision captures a 3D point cloud using the sensor
3. LHRobotics.Vision locates the container and some workpieces
4. LHRobotics.Vision calculates a collision-free removal path for a detected workpiece.
5. LHRobotics.Vision answers with the path points of the extraction path PROVIDE_OBJECT.
6. Further data such as an object side, a gripping point number or the fill level of the bin can be transmitted.
7. The cycle (incl. messages, status and point cloud) is logged and can be loaded later as an offline cycle and edited if required.

Note pose convention:

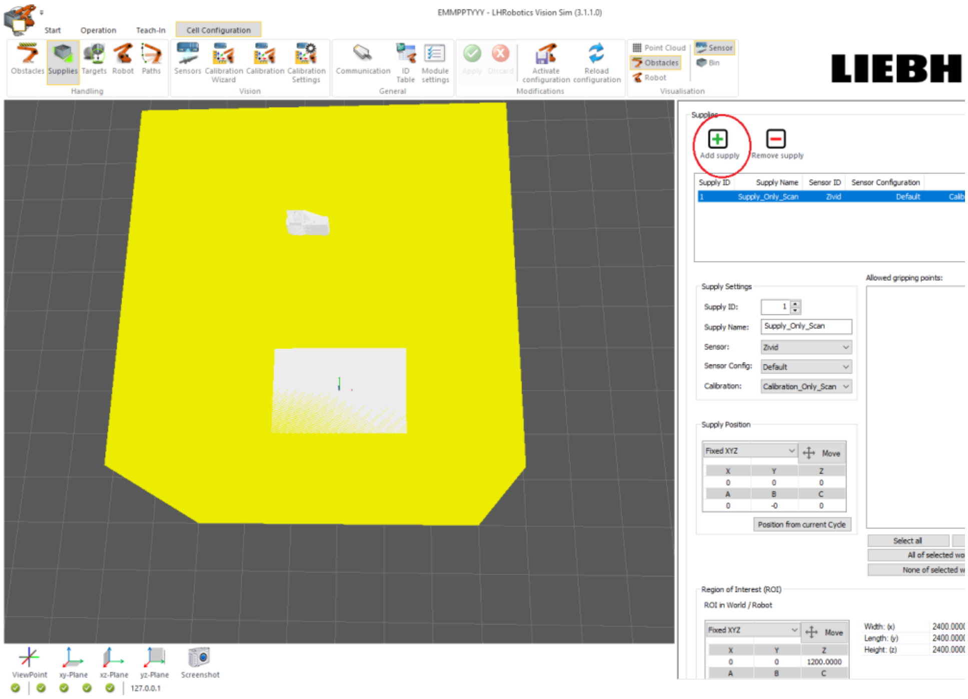
Kuka: EulerZYX
ABB: EulerZYX
Fanuc: FixedXYZ

5. Hand-eye-calibration

5.1 Preparation

5.1.1 Software

Each calibration is added to a certain location. This must be created under **Supplies** before calibration. For information on how to create supplies, see chapter 6.7 Creating a supply area.



5.1.2 Mechanics

How-to Video im LHRobotics.Vision Downloadportal

With stationary camera

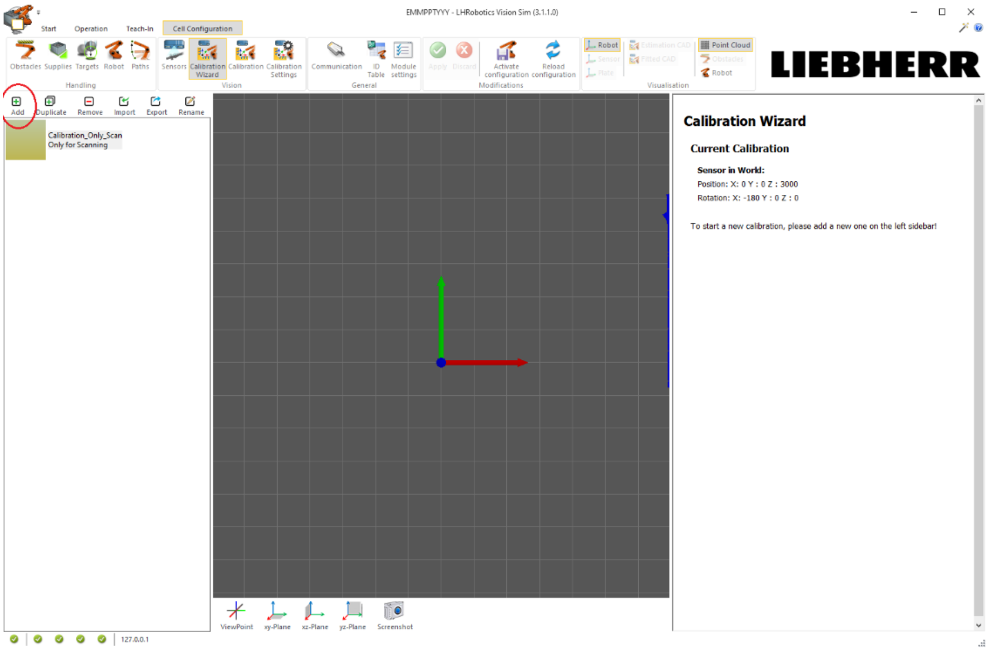
Mount the measuring stand with Liebherr-calibration-plate on the robot gripper. The calibration plate must be mounted on the last axis of the robot. This mounting position of the calibration plate must not change during the entire calibration process: Make sure that the set screws on the magnetic stand are tightened. The calibration plate must be completely visible to the camera.

For on-arm camera

The camera must be mounted on the last axis of the robot. The measuring stand is positioned on the floor, i.e. the scan location – without workpiece container – and positioned centrally below the camera.

5.2 Starting the calibration in the software

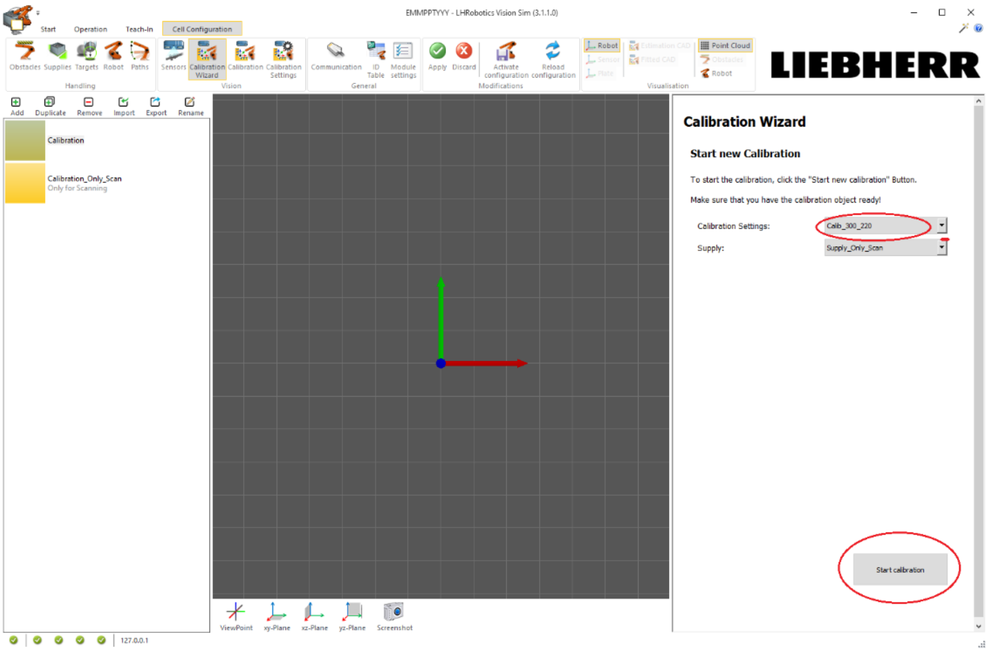
To create a new hand-eye-calibration, go **to Cell Configuration** and then to **Calibration Wizard**.



After adding a new calibration, the assistant can be started:

Attention: Make sure that you have selected the correct calibration plate and the created deployment location.

- This can be checked and changed in the menu item **Calibration settings**



5.3 Performing the calibration

Automatic calibration
As a rule, calibration is performed automatically, that means the robot moves to programmed positions. For this, there must be at least 30cm of space in the X, Y and Z directions. The plate must not be covered by the robot or gripper.
The customer must provide a robot program that first moves to a central position directly below the camera. The robot then receives the positions from LHRobotics.Vision, which it must approach one after the other for the calibration process. A camera scan is performed in each position.

With stationary camera: The robot moves relative to the camera.

Preparation	• Move robot to start position
Start	• Start calibration
Calibration	• Robot moves to 15 calculated poses • Camera scans in every position
Calibration done	• LHRobotics.Vision calculates calibration from poses and scan data

With on-arm camera: The plate lies on the floor (must not be moved). The robot moves to different scan positions and thus captures the plate from different angles.

Manual calibration
If automatic calibration is not possible, i.e. if the robot can travel less than 30cm in X, Y and Z direction, the magnetic stand with the calibration plate is positioned on the floor / scanning location (without the bin) below the camera. The plate must not move during this process. The robot must then move manually to 4 fixed points marked on the calibration plate.

Proceed as follows:

- Mount the measuring tip on the robot flange
- Align the measuring stand and set the calibration plate approximately horizontally to the floor.
- Move the robot manually to the holes of the calibration plate. Teach the position of the holes. Via the robot program a message must be sent to LHRobotics.Vision with the taught position.
- After teaching all holes, scan the calibration plate once via the **Calibration Wizard**.
- Then the robot must move to the holes based on the scan data.
- If the points are hit by the measuring tip, the calibration was successful.

Note: The accuracy of manual calibration depends largely on the teach accuracy and cannot be guaranteed. We recommend automatic calibration whenever possible.

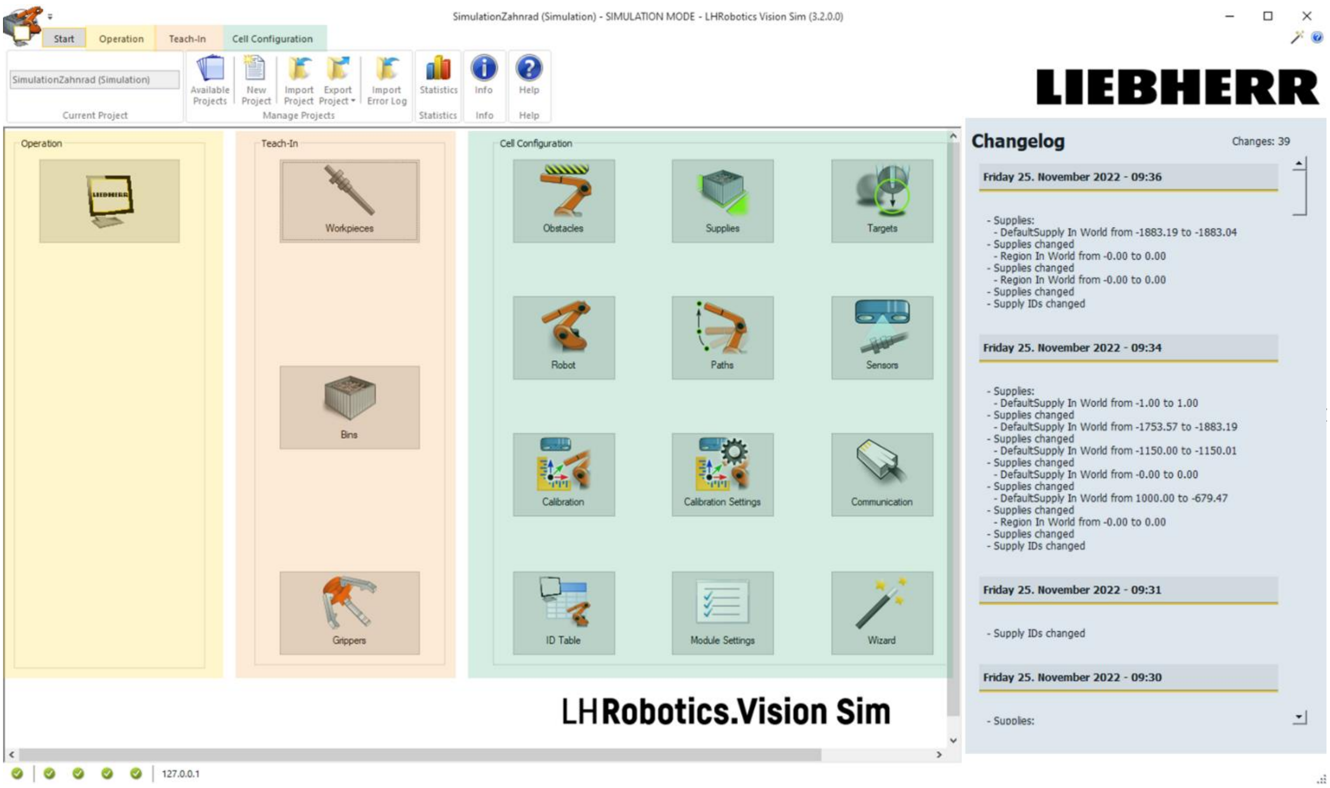
6. First steps in LHRobotics.Vision

6.1 General functions

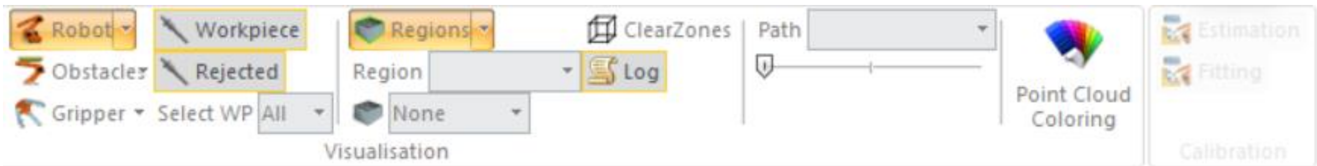
6.1.1 Graphical user interface GUI

The GUI allows you to edit the system settings and visualize the current operations. It is divided into the following categories: Operation (highlighted in yellow in the image), Teach-in (orange) and Cell Configuration (green).

In the start screen on the right (blue) is a changelog with all recent changes in the project.
The areas can be selected via the app icons or the tabs.

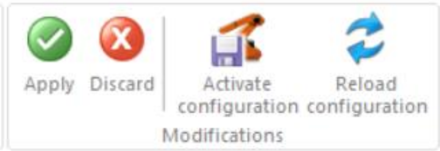


In areas with visualization, different components of the visualization, e.g. robots, obstacles (= the robot cell) or different regions can be shown or hidden in the ribbon.



6.1.2 Save

In order to edit settings in parallel with productive operation, there is a two-stage saving procedure:



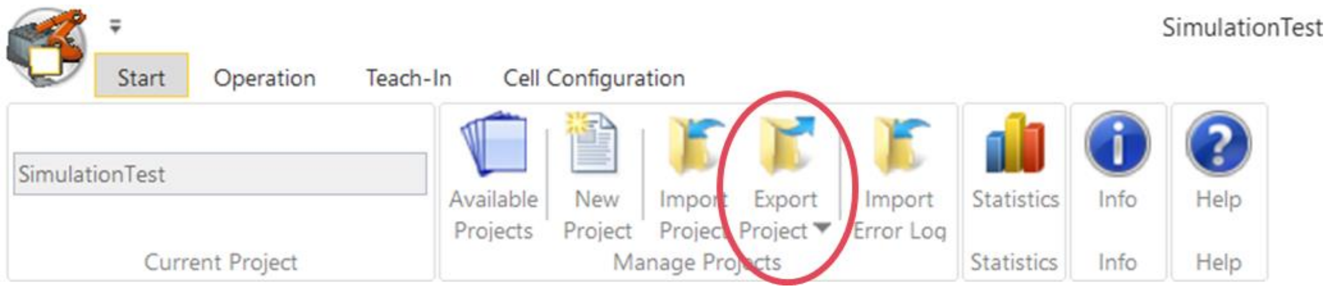
'Apply' = changes are only saved in the graphical interface
'Discard' = discard last changes

Only a click on 'Activate Configuration' applies the changes to the running operation. Now the changes become effective under **Operation**. Note that no requests may be sent from the robot during the saving process.

6.1.3 Save project

This can be used to save the entire project, including components, gripping points, etc.

On the home page, click on 'Export Project'.

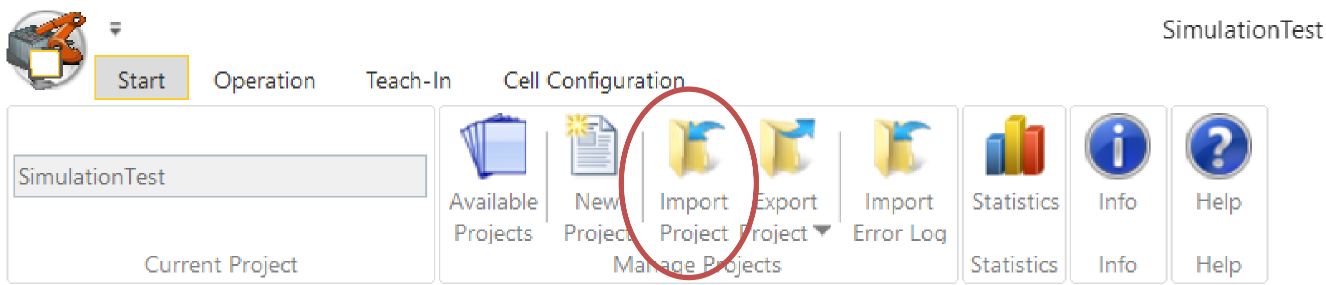


After that, select the location of the backup file.
Note: The backup should be done regularly and stored on an external disk.

6.1.4 Restore project

This can be used to restore or import a complete project.

On the home page on 'Import Project'.



After that select a previously created backup file.

6.1.5 Export cycles - e.g. for error analysis

In order to analyze cycles and send them e.g. to Liebherr Service, cycle data can be saved and reloaded later. When exporting the cycle, a file with the cycle data is created (cycleData.json) and a file with the point cloud (scan.json).
The original logged cycles can be imported and used for simulation.

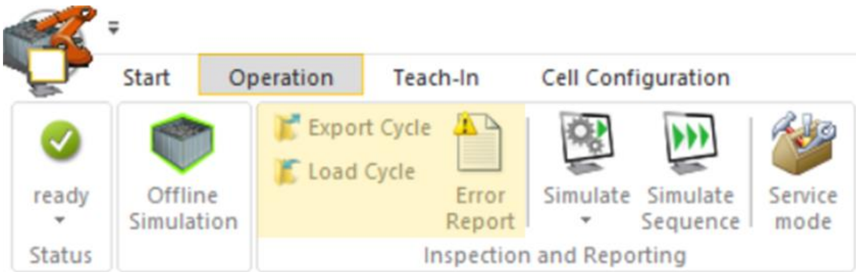
An erroneous scan can be saved via **Operation** > Export Cycle. All cycles with error messages are automatically saved in your LHRobotics.Vision path under projects > [project name] > log > [error name], e.g.

101_ProvideObject

103_NoSolutionFound

The files are loaded via **Operation** > **Import Cycle**.

Alternatively, an Error Report can be generated that contains a screenshot of the user interface and the project configuration file in addition to the cycle data:



For this purpose, a zip file is created that contains all necessary elements. In the LHRobotics.Vision path under projects > [project name] > error_report > [current date].

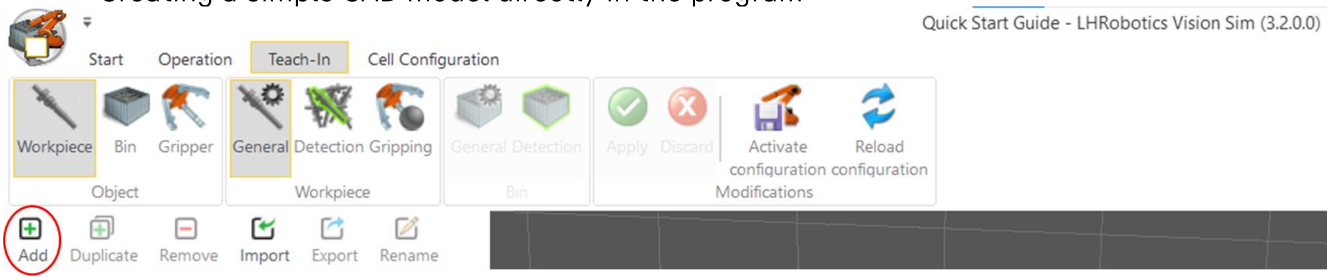
This is useful for failure analysis on another computer or when contacting our service. The error report is imported on the start page via ‘Import Error Log’.

6.2 Creating a workpiece

6.2.1 Create CAD model and workpiece parameters

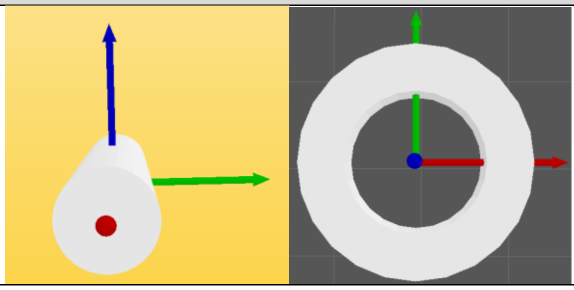
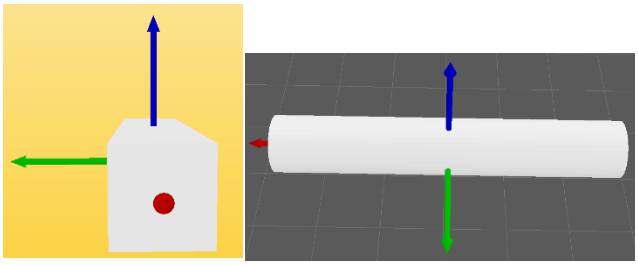
A workpiece can be created via **Teach-In** > Workpieces by 2 different methods:

- Import of a CAD model (.stl or .wrl).
  Attention: A too complex 3D model slows down the application - a click on 'Simplify' optimizes your model
- Creating a simple CAD model directly in the program

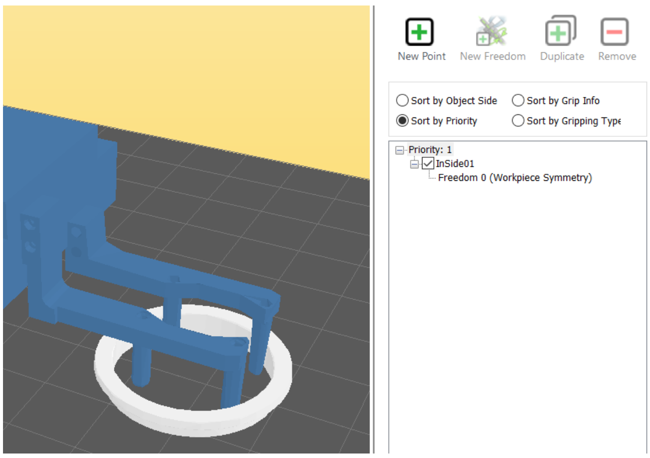


Define one or more rotational symmetries in the 'Symmetries' area. This...

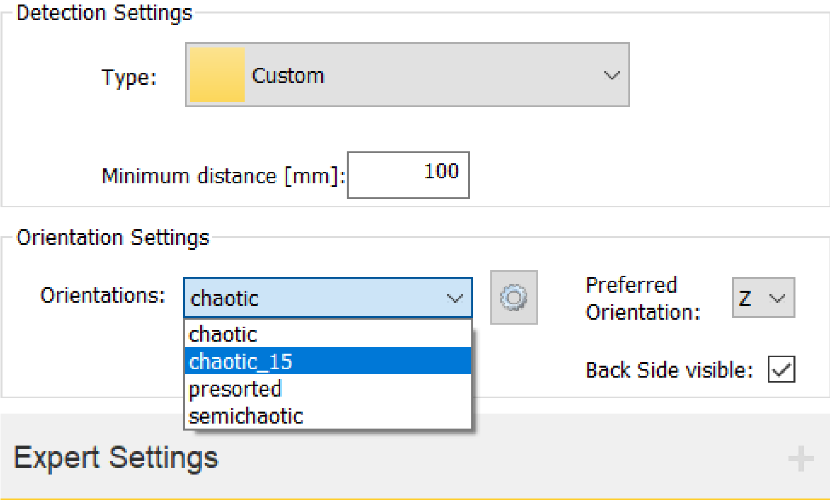
- ... speeds up the calculation
- ... is mandatory for the automatic teach-in of the search parameters (available from software version 3.3)

Continuous	Discreet
	
e.g. Continuous rotation around the X-axis (red) or Z-axis (blue)	e.g. 4-fold discrete rotation around the X-axis (red), 2-fold discrete rotation around the Y-axis (green)

You can check the symmetries as soon as you create gripping points: Here, the degrees of freedom are automatically created based on the workpiece symmetries. A degree of freedom is created for each stored workpiece symmetry:



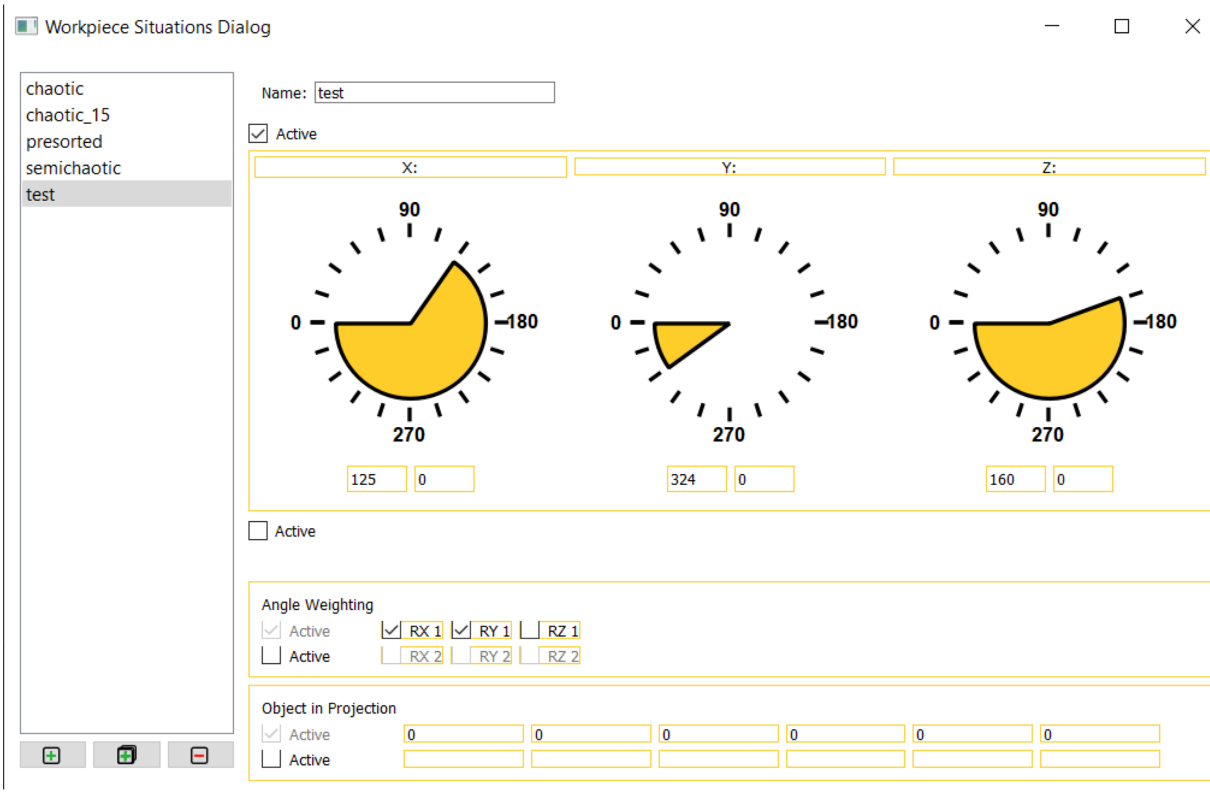
The orientation of the workpiece is then defined under Teach-In > **Detection** > Orientation Settings.



These inputs determine which workpiece orientations in space the software expects and calculates accordingly.

If a workpiece is newly taught and accepted into the configuration ("Activate configuration"), the system calculates the poses that the workpiece can assume. In case of chaotic orientation, this may take some time.

You can enter your own orientations by clicking on the cogwheel to the right of the dropdown:



The larger the degrees of freedom (orange area), the more complex is the calculation & requires more time. Therefore, always specify only the orientations that you actually expect.

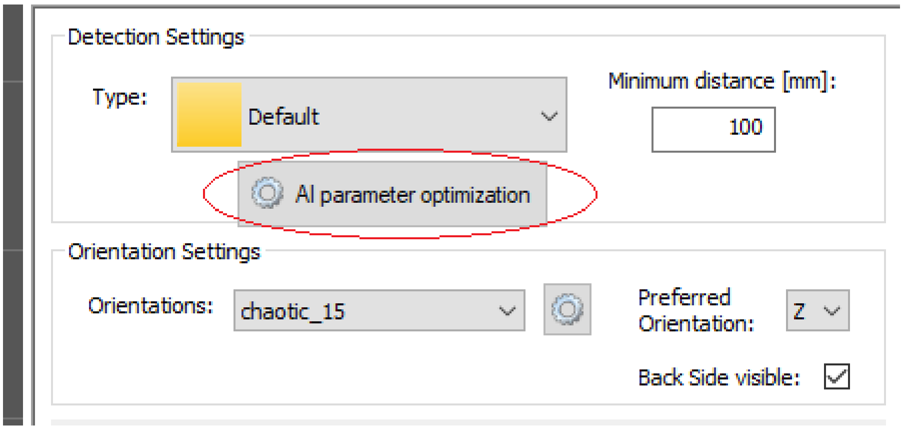
6.2.2 Automatic teaching of the detection parameters using AI

In order to accelerate the teach-in of new components, there is a new feature in LHRobotics.Vision as of version 3.4 to determine the optimal detection parameters by means of an AI algorithm.

The following must be done before the automatic teach-in can be started:

- The sensor must be created, calibrated and set (detection settings in .yml file see chapter 3.3). The settings in the .yml file must not be changed afterwards, otherwise the teach-in process must be repeated.
- The box must be created and set
  - Please make sure beforehand that the system can detect the box. To create a box, see chapter 6.3 Creating a workpiece container. If the box is not detected, refer to chapter 8.2.
- The symmetries of the workpieces must be correctly parameterized
- The orientation that the component may have must be set, see chapter 6.2.1

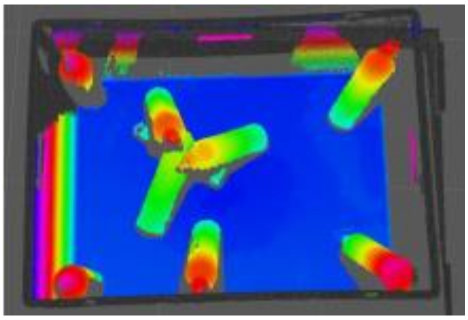
Start the parameterization under Teach-In > **Detection** > Orientation Settings



In the next dialog window, select the box and the box location / supply.

Fill real box with parts

At least one box situation should be produced in which the workpieces lie as they are expected to appear later. **Important:** All possible orientations should be covered!



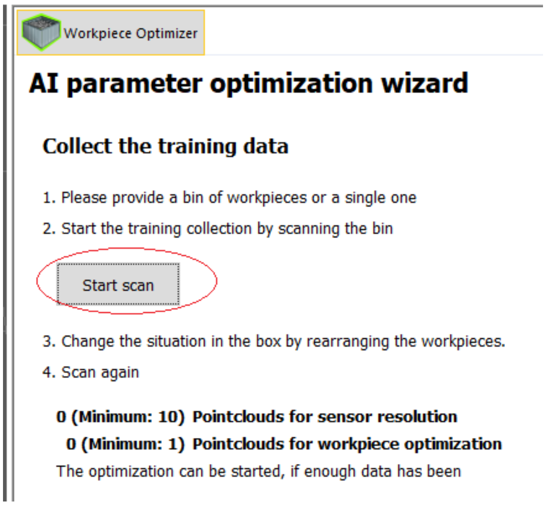
If there are too few workpieces to cover all orientations, please create two or more scans with the different orientations (i.e. perform the next described step – recording the training data – two or more times).

Starting a new training cycle = recording the training data

A new cycle is recorded via the "Start Scan" button. During the first cycle, 10 scans are automatically recorded for the analysis of the sensor.

Important: The box must not be changed or moved during the 10 scans!

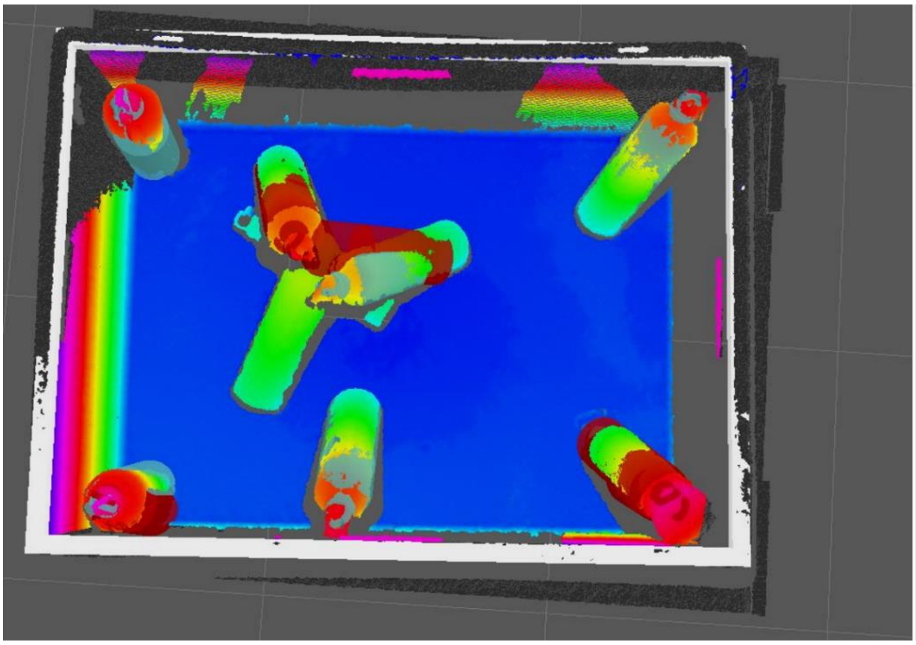
One of these 10 cycles is automatically used as the cycle for workpiece optimization. After that, another cycle for workpiece optimization can be recorded with each additional click on "Start Scan". If at least one box situation has been recorded, please click on next to enter the next dialog.

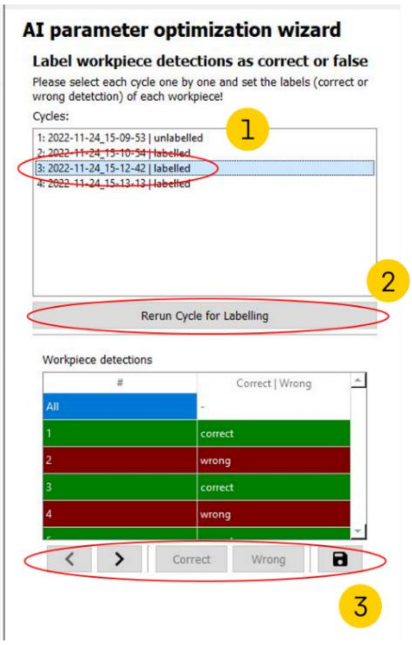


Training data generation

For the algorithm to work, the system needs to be told which workpieces were correctly and which were falsely detected. For this purpose, LHRobotics.Vision makes a rough initial estimate and declares as many elements as possible as detected workpieces. These must now be sorted out by the operator.

1. Select a cycle in the user interface
2. Start the rough detection of the selected cycle.
3. Generate the training data → Mark the workpieces as correct (if correctly detected) or wrong (if an element was detected as a part but actually isn't). To do this, consult the 3D view and click on the corresponding "Correct" or "Wrong" button.
4. If some parts were not detected, create another cycle with more parts in the orientation that was not found.

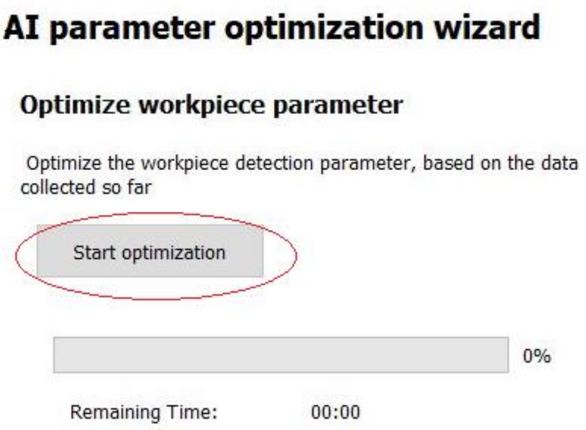




After all training data has been created, go to the next dialog box.
There you start the optimization.

Now, the detection parameters are determined automatically. This may take some time depending on the complexity of the workpiece and the number of training cycles.

Let the process run through completely or end it manually when all components have been detected correctly and without false detections.



6.2.3 Search for workpiece sections using segmentation

In situations where the workpiece is only partly visible, it is possible to search for segments. For each segment different recognition settings can be defined. By clicking on ‘Extract’ the CAD can be segmented into different segments.
Depending on the selection in the dropdown...

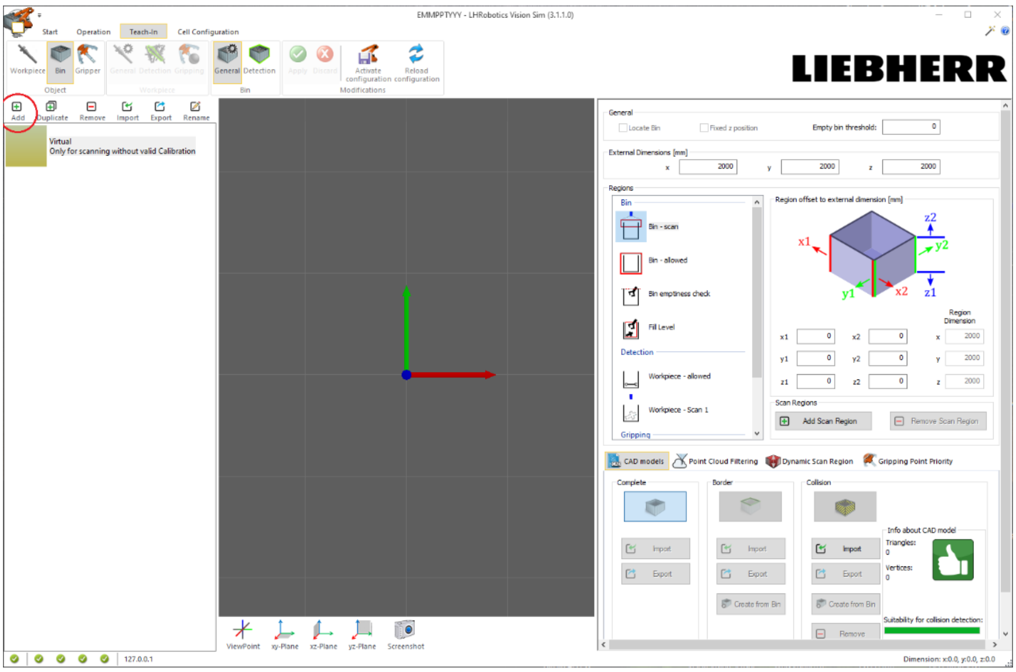
- the workpiece is segmented along the xy and xz plane and you get 4 segments.
- By selecting ‘Connecting rod’ the workpiece is segmented into 4 segments along the yz-plane.
- By selecting the original workpiece and clicking on ‘Integrate’, all segments can be integrated into a single workpiece. The recognition settings defined for each segment are retained even after the segments have been integrated into one.

For more information, use the online help in the program.

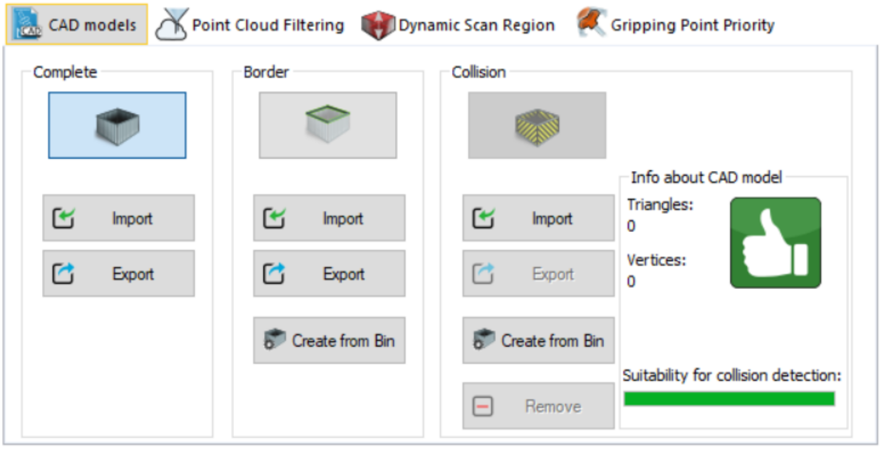
6.3 Creating a workpiece container

A box can be created via **Cell Configuration** > Bin by 2 different procedures:

- Import of a CAD-model (.stl or .wrl)
- Creation of a simple CAD-model directly in the program



In the area **Regions** you can define different areas of search:
– Bin – scan = area in which the workpiece container is searched
– Bin – allowed = area in which the workpiece container may be located
– Bin emptiness check / fill level = area in which workpieces are searched to determine the emptiness level
– Workpiece allowed / scan = range in which the workpieces can be respective to the border of the container. For example, if the crate has a 20mm border, 20mm can be entered in the X and Y directions.
– Collision (border) = here a kind of safety distance can be entered, so that collisions can be omitted. The gripper and robot keep away from the edge of the crate by this value.

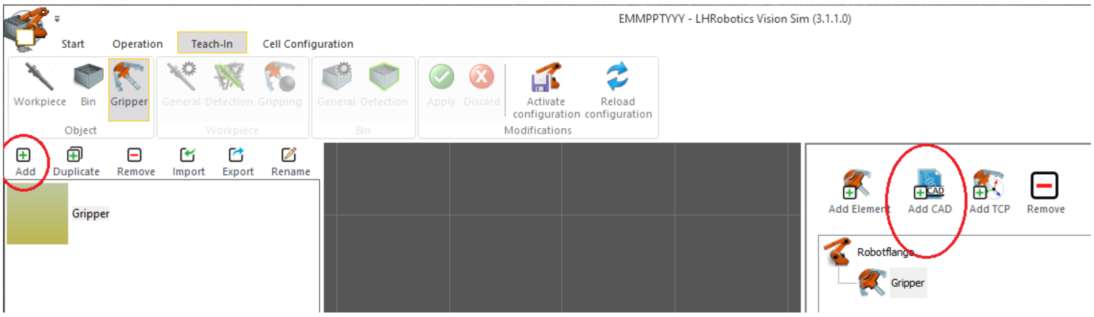


It can be useful to have different CAD models, one complete (for visualization), one with only the edge of the box (for localization), and possibly one more to check for collisions when looking for gripping points.

For more information, use the online help in the program.

6.4 Creating a gripper

In the list on the left, a new gripper must be created. On the right side, any number of CAD models can then be added to this gripper.



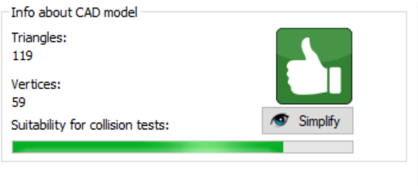
The following elements are mandatory:
– At least 1 CAD-model
– At least 1 TCP for creating the gripping points

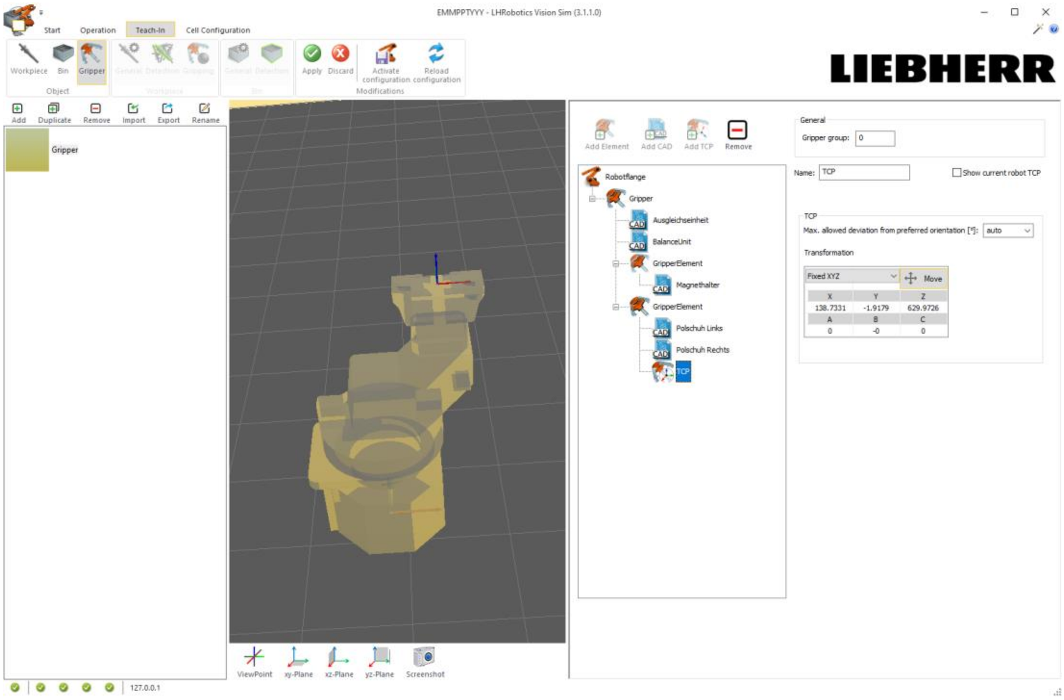
Note:
It is advantageous that the individual components of the gripper are each created as a single CAD model. This has the following advantages:

- The calculation time for the collision check is reduced
- A certain number of allowed collision points with the point cloud can be set for each component
- Components can be simplified directly in the program

The thumb on the right should be green for each component, otherwise the calculation can slow down considerably.

For more information, use the online help in the program.

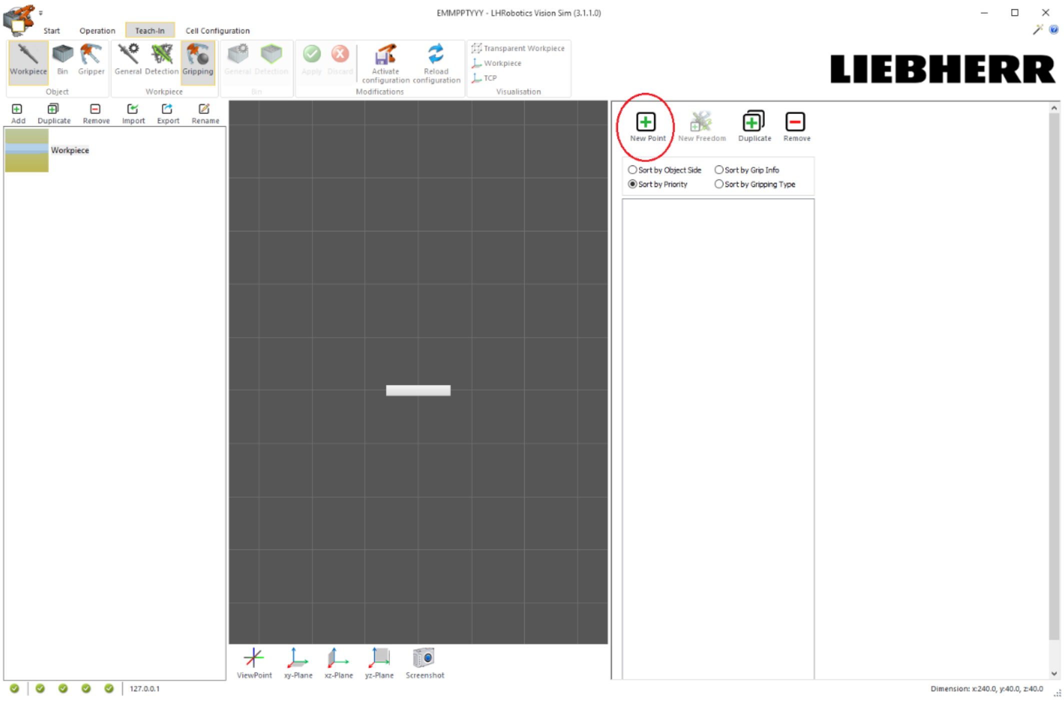




6.5 Creating gripping points

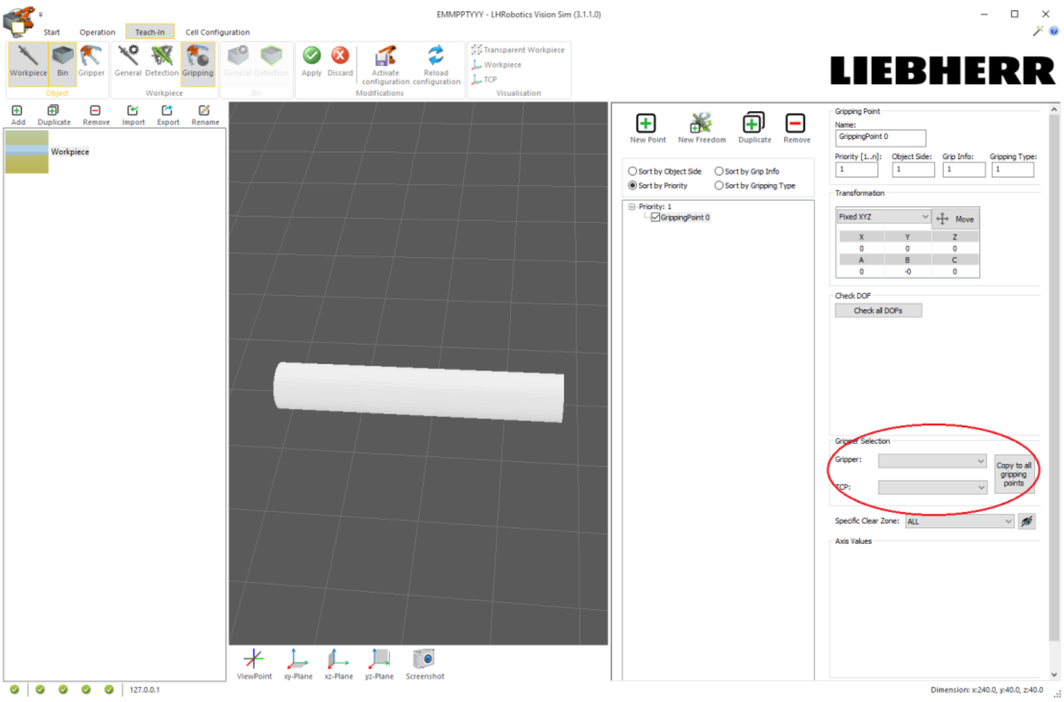
If a workpiece and a gripper have been created, the gripping points can now be added. To do this, select under **Teach-In** > Workpieces and then switch to the **Gripping** menu.

The workpiece can be selected in the list on the left. On the right side, any number of gripping points can be created.



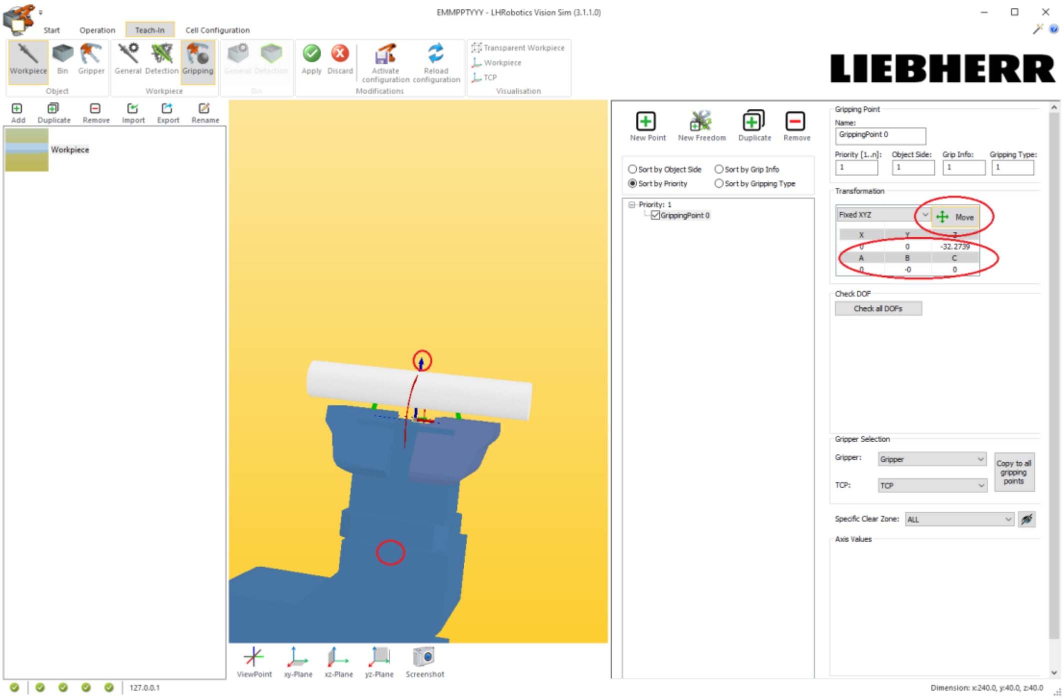
Note:
Many gripping points (>500) can have a negative effect on cycle time!

The appropriate gripper must be selected for the created gripping point.



The gripper can be positioned anywhere on a workpiece using the mouse.

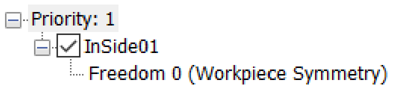
- Keep left mouse button pushed
- Mouse on the gripper to move it completely
- Click on a coordinate axis to move it only in one e.g. the X-direction



Alternatively and/or for fine positioning, the coordinates can be entered manually.

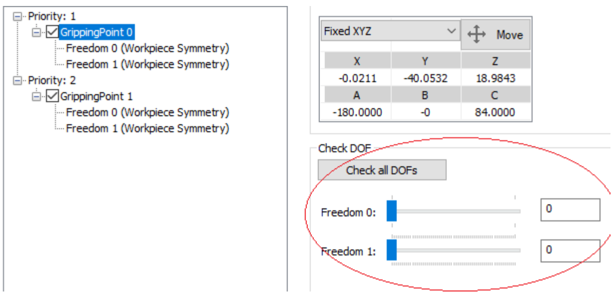
6.5.1 Degrees of freedom

For each gripping point degrees of freedom – DOF – can be created:
For rotationally symmetrical parts, the axis of rotation is automatically created as one DOF.



The gripper may here e.g. dock at the gripping point, no matter in which orientation the X-axis is located.
DOF can be defined ‘continuous’ or ‘discrete’ in freely definable steps. In the latter case, the number of steps must be given.

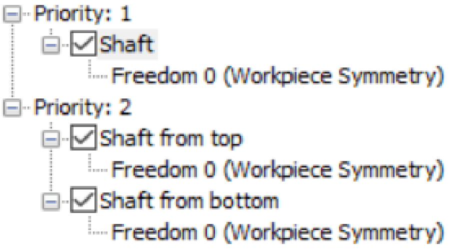
To check the set degrees of freedom there is a slider for each degree of freedom.



6.5.2 Priorities

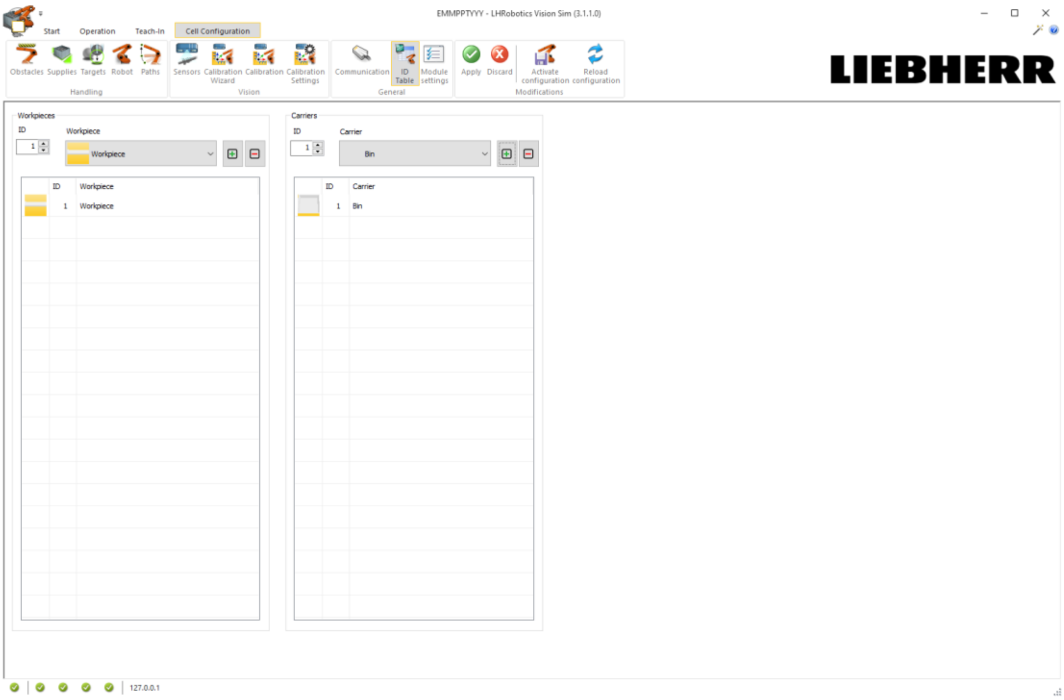
By assigning priorities, you can e.g. preferably use more reliable gripping points. The gripping point with the highest priority is preferred.

In this example, the gripping points in the group with Priority 2 would be preferred:



6.6 Creating IDs for workpieces and containers

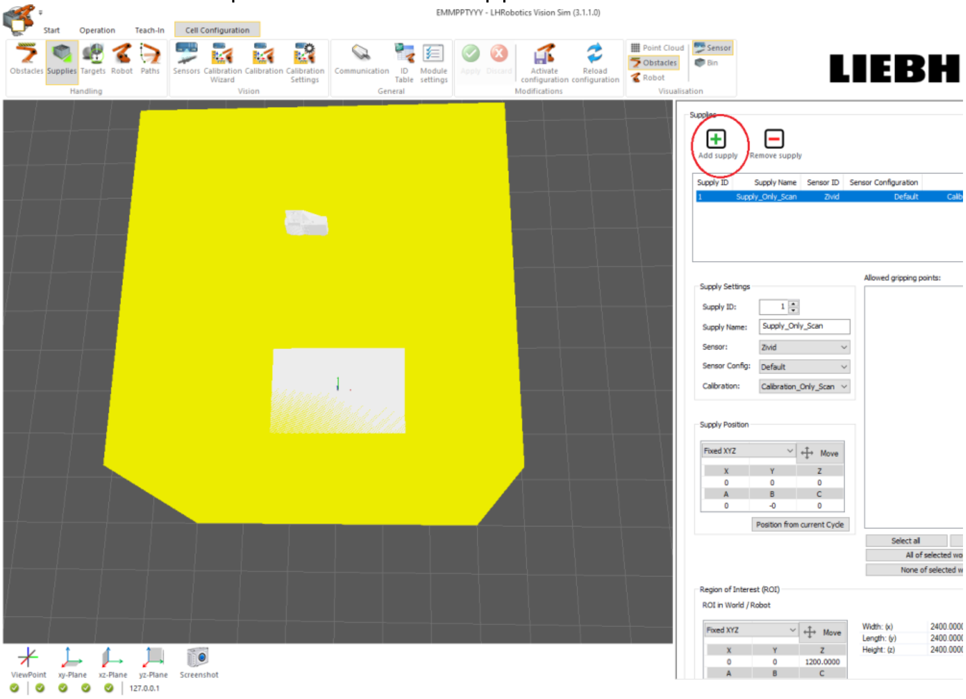
In order for the robot to be able to request which part is to be taken from which box, an ID must be assigned to each created component or box, which the robot then requests from the LHRobotics.Vision software.



Important: Each workpiece and each box require a unique ID.

6.7 Creating a supply area

The areas in which workpieces are to be supplied and scanned are created under **Supplies**.



Assign a name and select the sensor and a configuration and camera calibration under ‘Settings’.

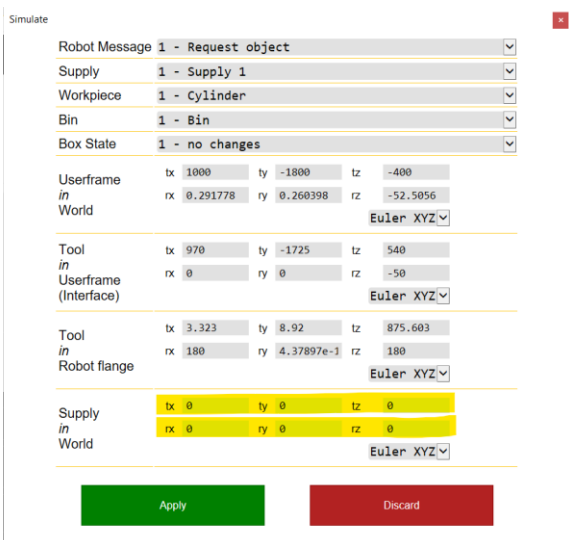
- Sensor = the camera that will be used at the supply location.
- Sensor Parameter = parameters from the camera SDK (e.g. Zivid Studio or Ensenso NXView)
- Calibration = the hand-eye-calibration

A separate hand-eye calibration must be performed for each supply position.

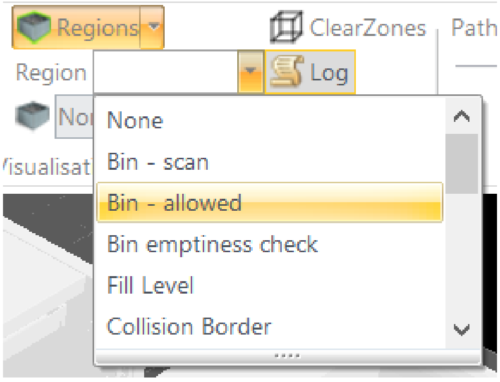
The **Supply Position** is the area within which the workpieces are supplied. ‘Position from current Cycle’ generates a suggestion for this area from the last camera scan. It can also be defined manually by entering the coordinates.

Define ROI = Region of interest = the area relevant for the camera scan. By defining a ROI, the acquired 3D point cloud is tailored to this region. Everything outside of it will be ignored for container detection, workpiece detection and gripping calculation.

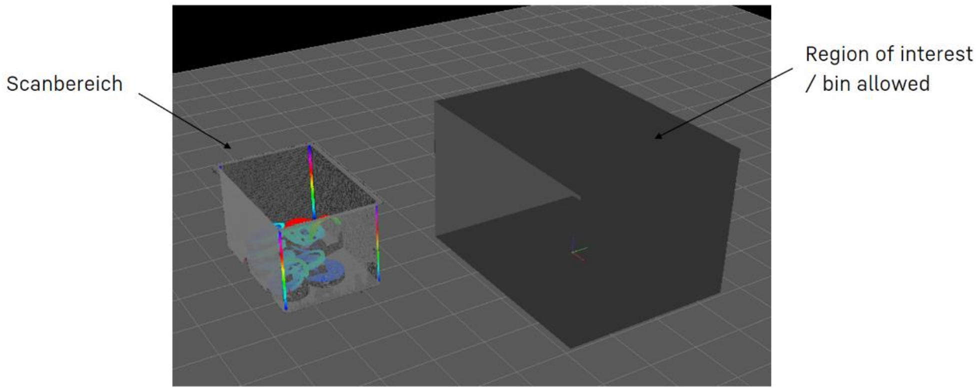
After setup, check the ROI and Supply Position: Create a scan via **Operation** > **Simulate**. Also check the coordinates under ‘Supply in world’ for correctness:



Then show the ‘Bin allowed’ area in the **Regions** dropdown menu:



In the 3D view you can then check if the scan area (here you see the camera image of your parts) and the ROI (shown as a gray box) match.



If this does not match (as shown in the picture), correct it via the coordinates or under **Supplies** > **Region of Interest** and the button ‘Create ROI according to latest scan’.

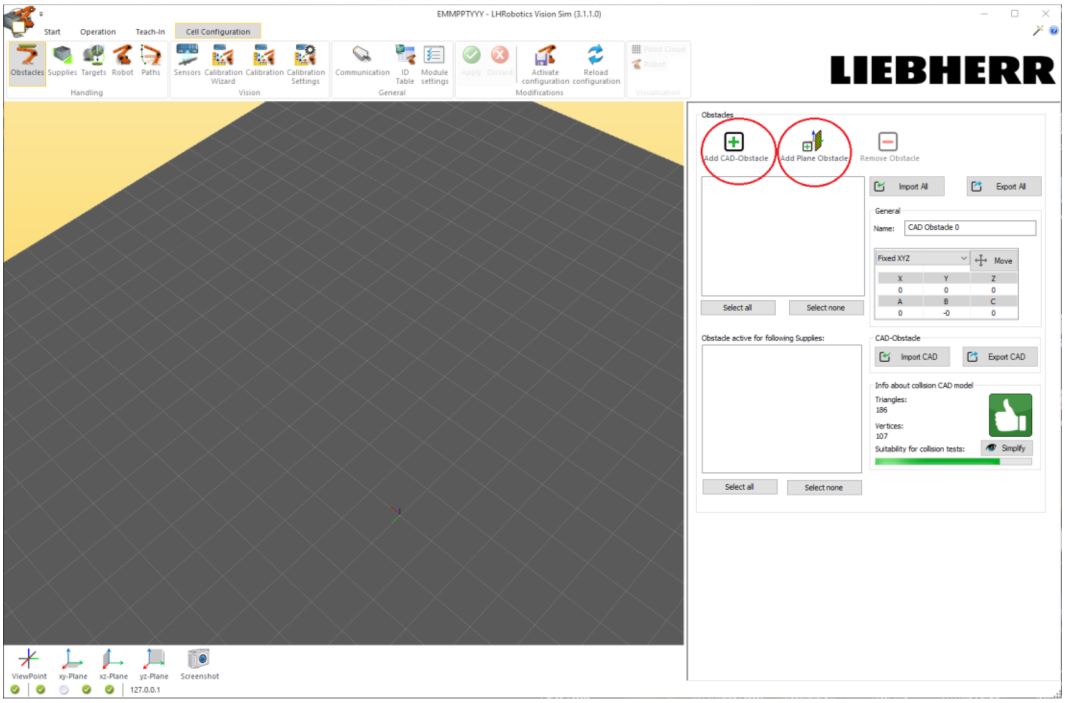
6.8 Creating the cell (optional for pro-version)

Cell Configuration > **Obstacles**

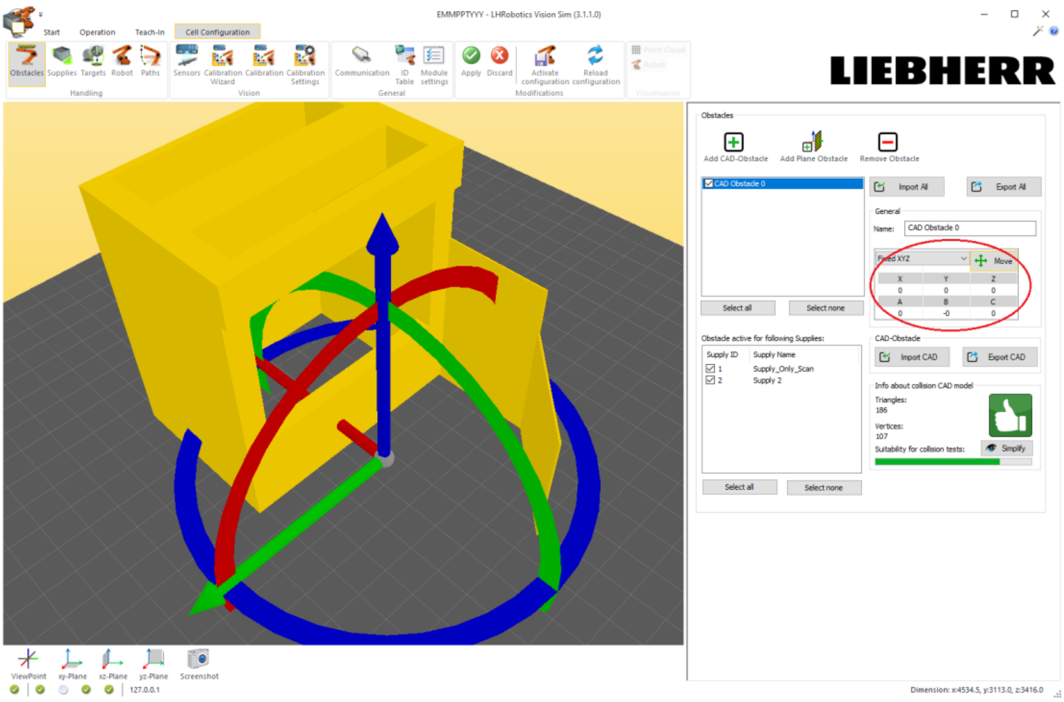
Creating the cell is optional and not available in the LHRobotics.Vision Basic license.
The purpose of implementing the cell is to avoid collisions with the robot and the gripper.

A component can be created via 2 different options:

- CAD-model (.stl or .wrl)
- Adding a plane



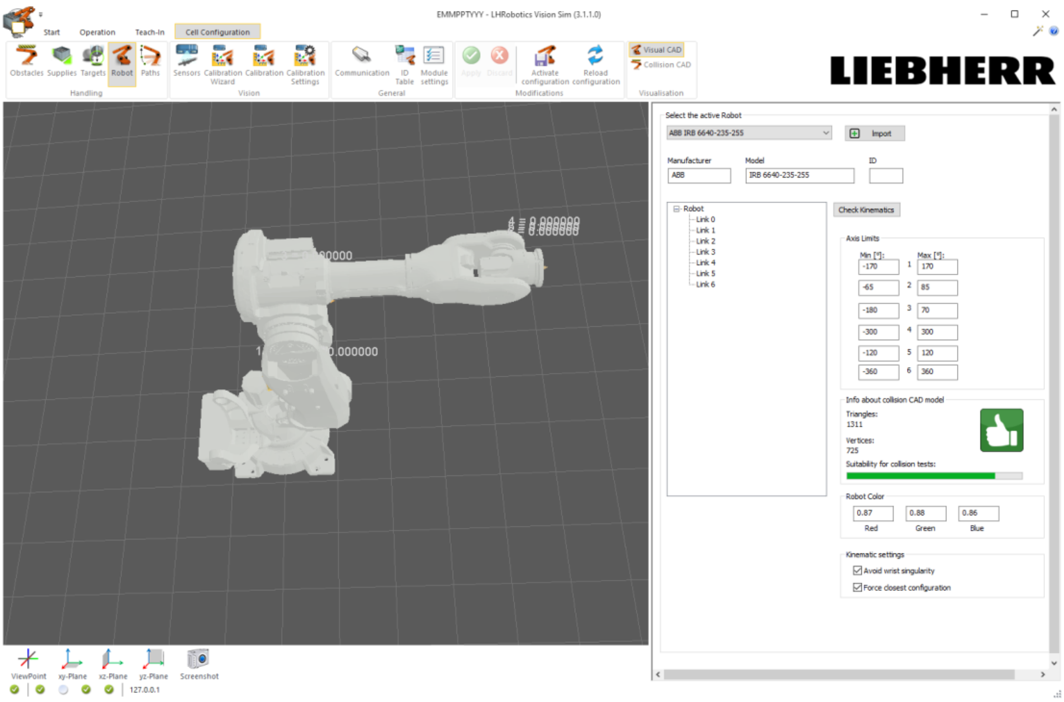
The model/ plane can be placed in the 3D scene as desired afterwards.



6.9 Selecting the robot (optional for pro-version)

Cell Configuration > Robot

Robot selection is not available in LHRobotics.Vision Basic license.



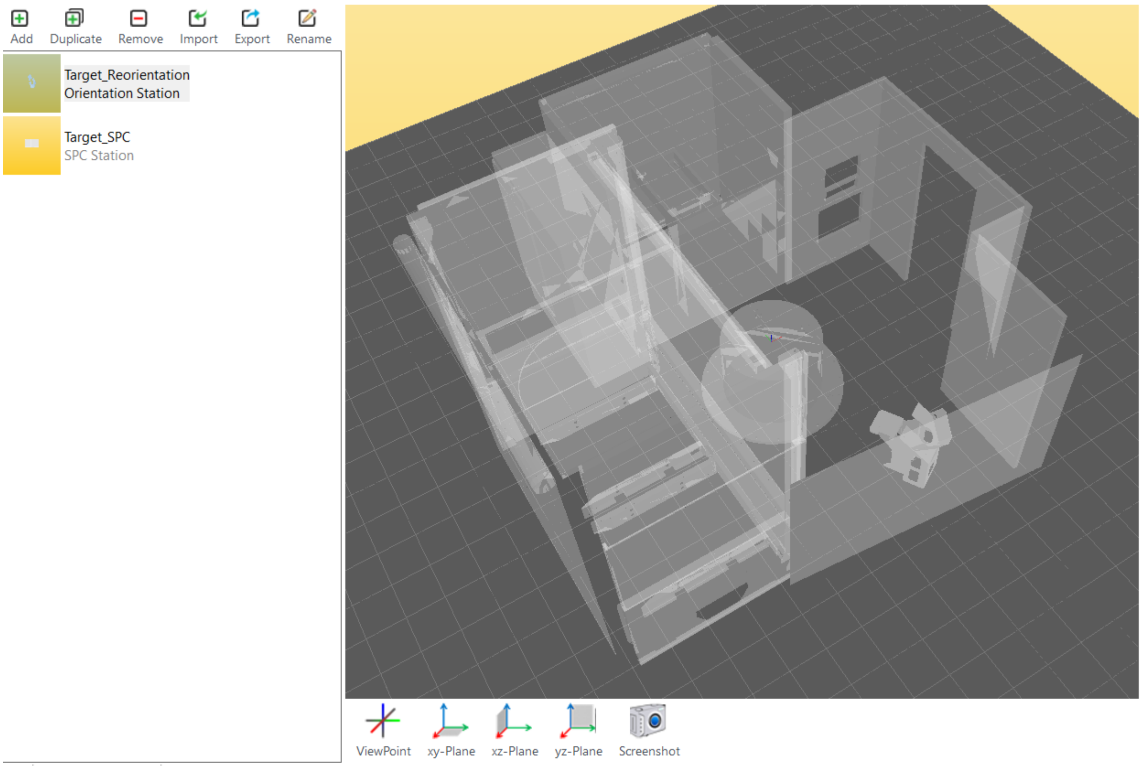
On the right side you can select the robot you need.

You can also define your axis angles here so that no path is generated that cannot be reached by the real robot.

If the required robot is not available, please contact Liebherr Service.

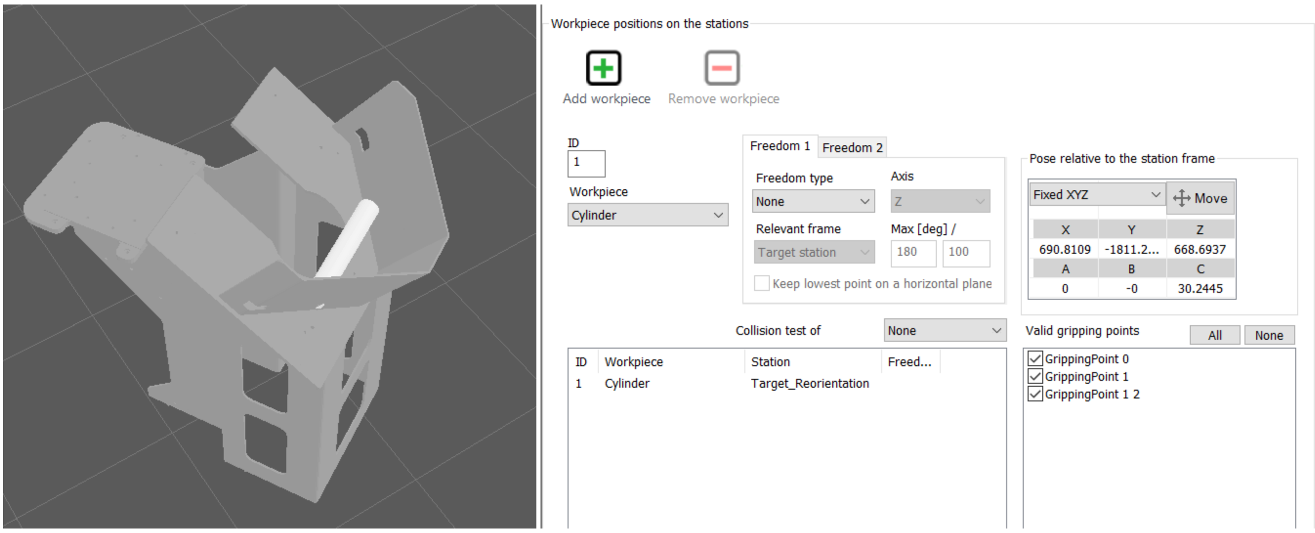
6.10 Selecting a target location (optional for pro-version)

A deposit location can be added under **Cell Configuration** > **Targets**. For this purpose a 3D model is imported. Each target receives its own ID, which is addressed in the robot program.



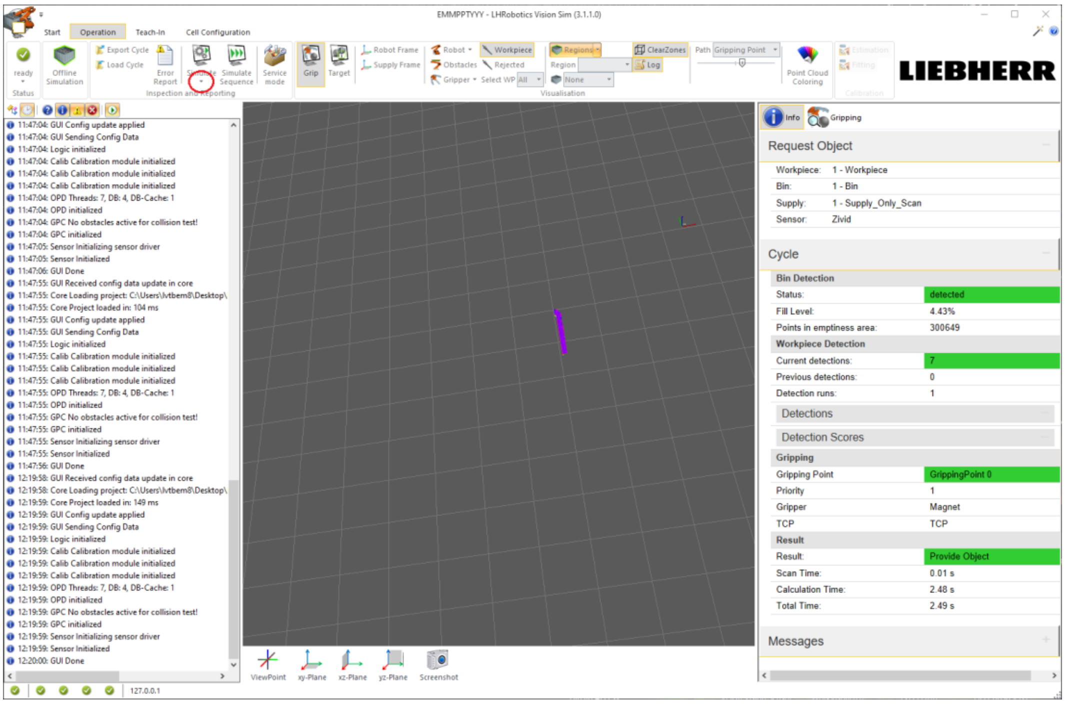
The deposit station can be positioned via the coordinates under Default Station Frame.

Then the workpieces to be positioned on the respective deposit location are added. The workpiece is selected under the dropdown **Workpiece** (all workpieces created in the respective project can be selected), an ID is entered and positioned via the coordinates.

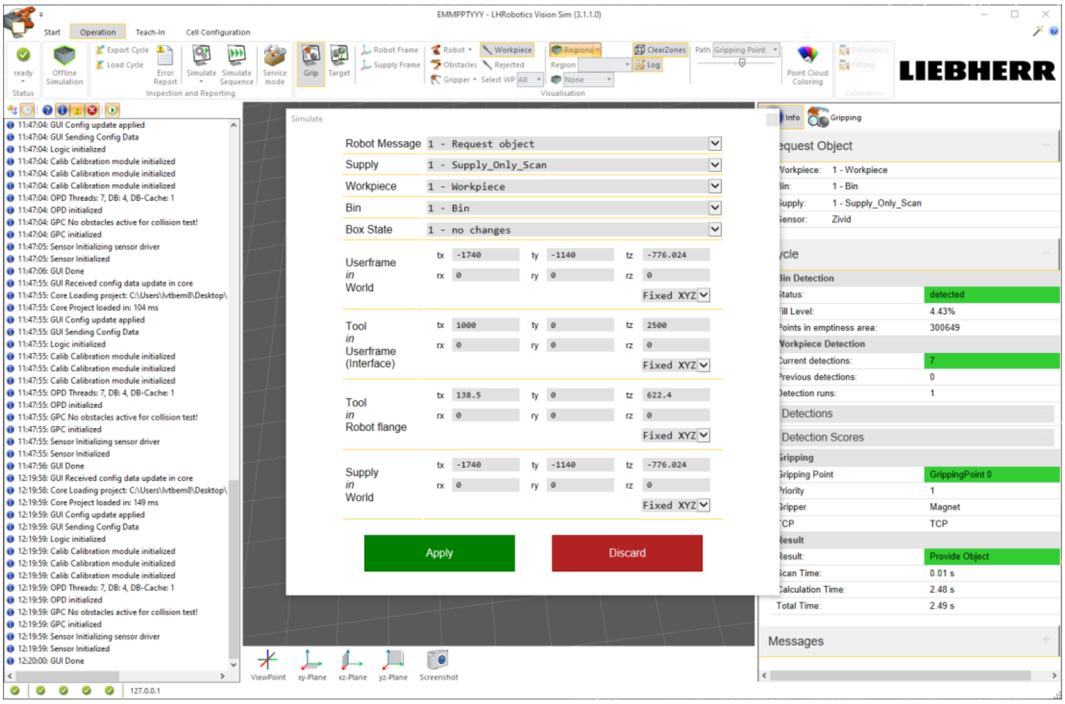


6.11 Simulation of the scanning process

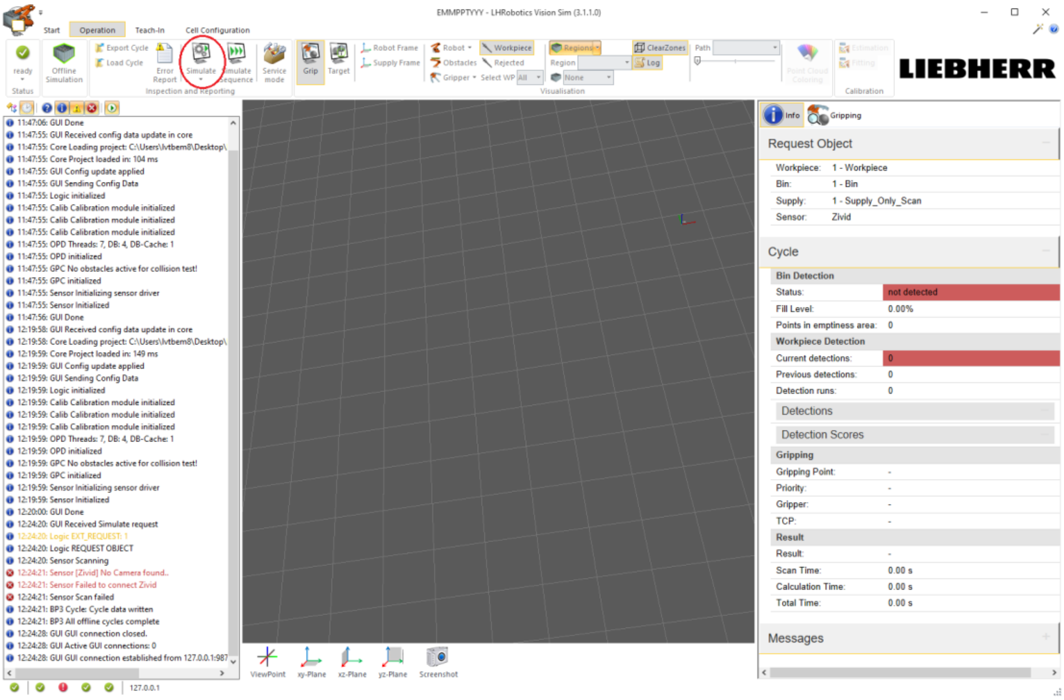
To check if everything has been set up correctly, the whole process can be simulated. To do this, switch to the **Operation** tab.



Select robot message 1 and the removal location, the component and the box which are to be detected.



Then the process can be started.

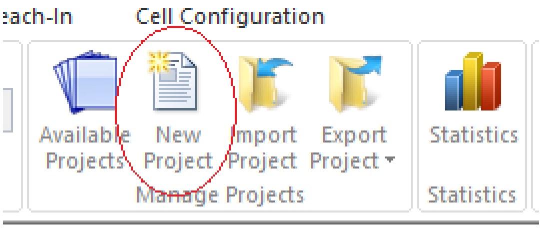


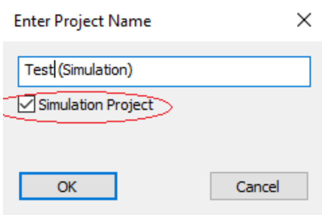
7. Simulation (Option 'Sim-Plugin")

7.1 Setting up a simulation project

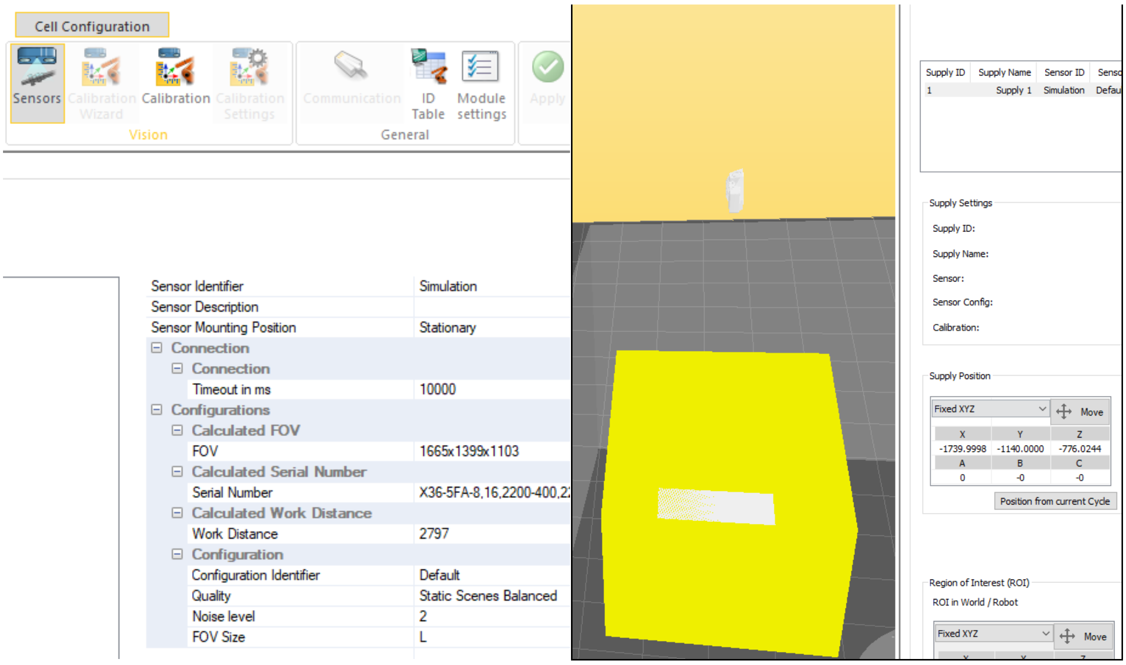
If the LHRobotics.Vision Sim plugin is to be used, a separate simulation project must be created. Unlike normal projects, certain parameters are set as unchangeable so that the simulation works.

1. Create a new simulation project. To do this, click on 'New Project' in the main menu:

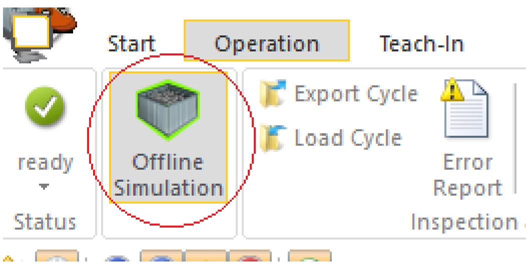




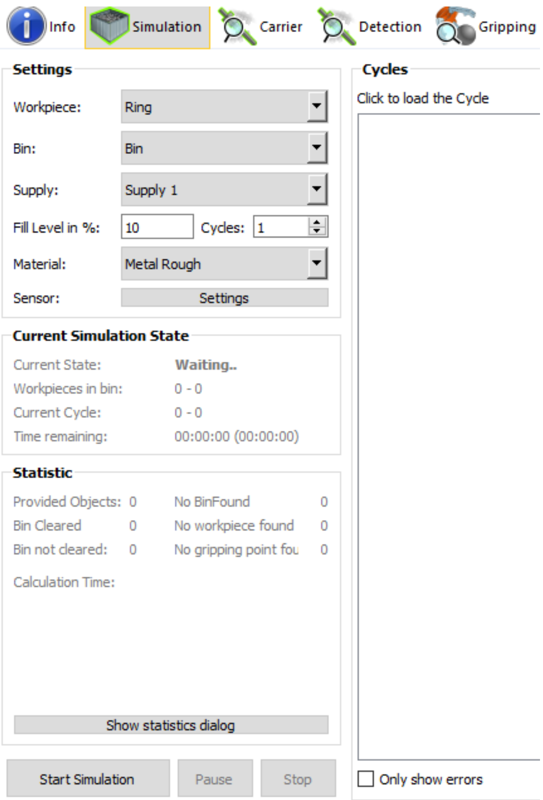
2. At Cell **Configuration > Sensors** resp. **Cell Configuration > Supplies** you can adjust parameters for the simulation sensor resp. change the bin position.



3. Click on 'Operation' and there on 'Offline Simulation

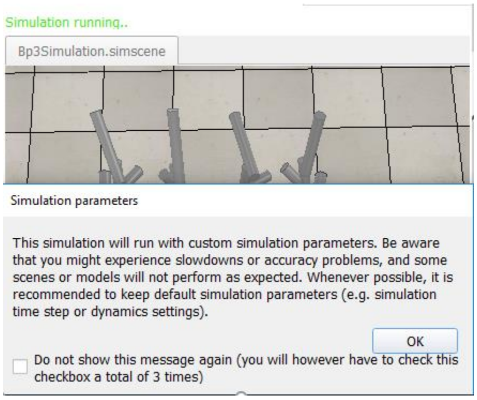


4. The configuration dialog now opens in the right-hand window.



Here you can choose which part should be searched for or to what degree the box should be filled. The simulation is also started here. In the lower part you can see the simulation environment and how the parts fall into the crate.

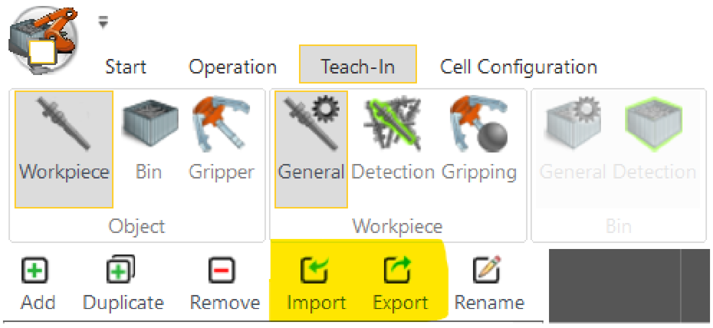
At the first successful start of the offline simulation the following dialog appears. This must be confirmed a total of 3 times and will then no longer appear.



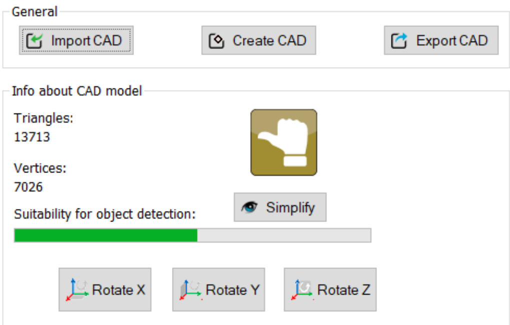
The workpiece that you want to simulate must be created in the simulation project. If it is not, switch to the project in which the workpiece was created via **Start > Available Projects** and export the workpiece via **Teach-In > Workpiece** and 'Export'. Gripping points are exported as well. Import the workpiece into your simulation project.

Remember to also add the new workpiece to the ID Table after the import. Only workpieces that are stored in the ID Table can be selected for the simulation.

The gripper must also be transferred to the simulation project via **Teach-In > Gripper** and ‘Export’.



Attention! Gripper and workpiece should not be too complex, otherwise the calculation times will increase. Pay attention to the messahes by the software. An example of a workpiece that is too complex:



7.2 Carry out and analyze simulation

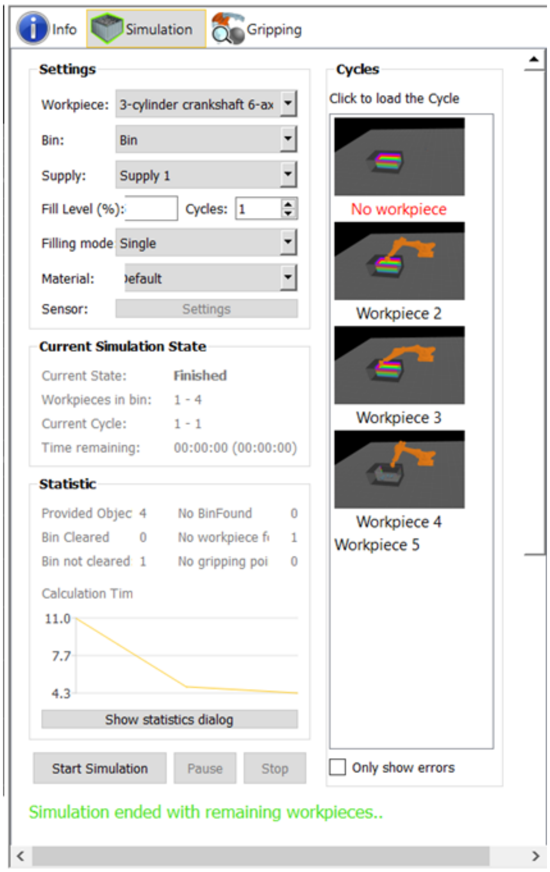
Under **Operation > Offline Simulation** > Settings, select the workpiece, the workpiece carrier and the supply and enter a filling level in percent.

Select the filling mode ‘Single’ and the material ‘Default’.
Then start the simulation.

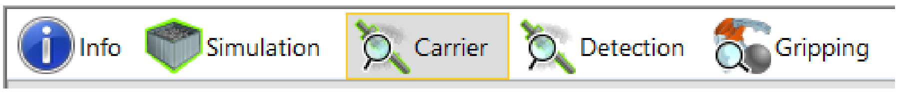
If the parts fall through the box during the simulation, for example, or if the simulation is too slow, please select CAD Reducing > ‘QGull’.

During the simulation, a CPU utilization of up to 100% can occur. This is no problem, but a desired status so that the maximum computing performance is available for the calculation.

For each scan a cycle is created, which shows the situation incl. robot position by clicking on the graphic in the list on the right.



Once you have selected a cycle, the ‘Carrier’, ‘Detection’ and ‘Gripping’ tabs appear.



Under ‘Carrier’ and ‘Detection’ you can see how the algorithm for part detection proceeded and which poses were calculated. This allows conclusions about improvements, see Chapter 8.2.

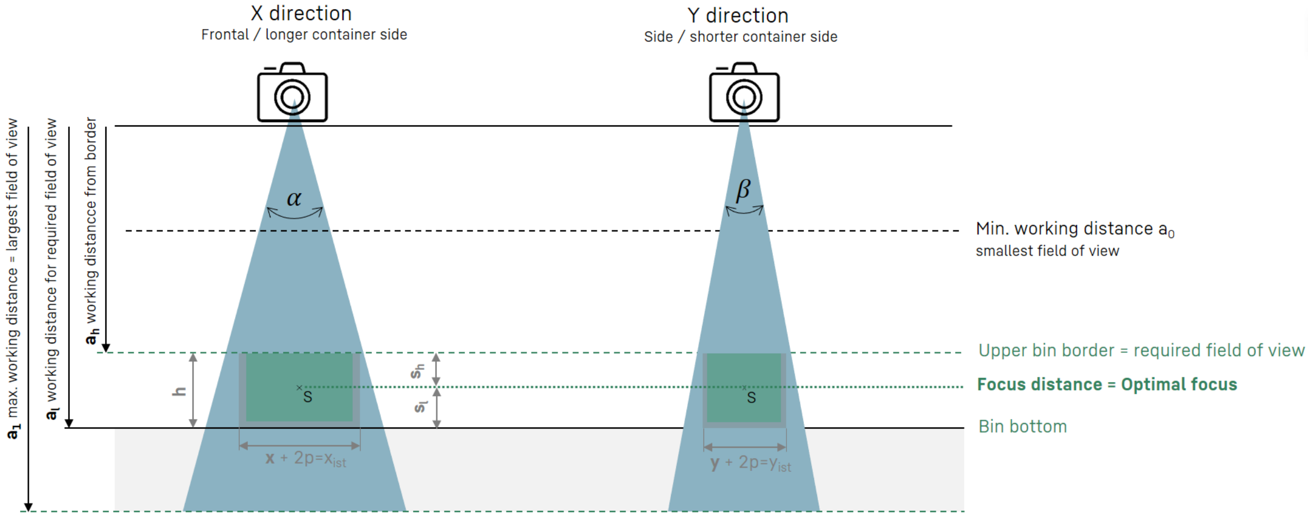
In the ‘Gripping’ tab, you can see the reasons why a gripping point was not possible, e.g. collision with the workpiece container.

8. Troubleshooting: Workpiece detection

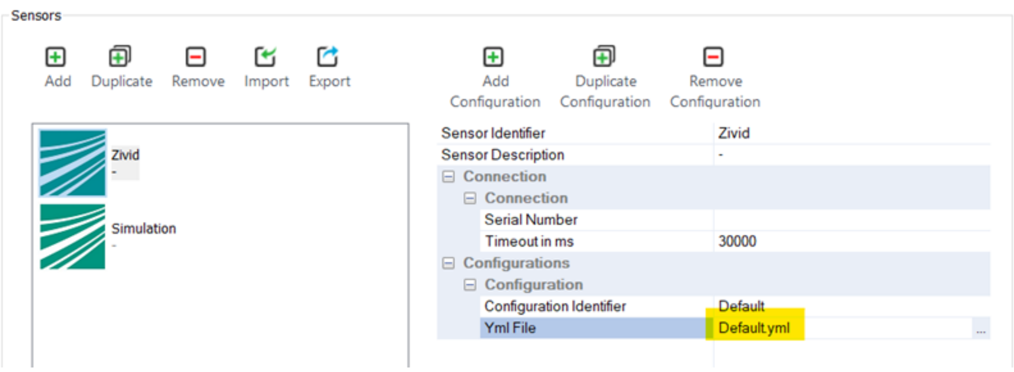
8.1 No workpieces were found -check point cloud

If problems with workpiece recognition occur, it is first recommended to check the point cloud. Workpieces can only be reliably detected if the point cloud is sufficiently accurate.

- Check the working and focus distance using the data sheet of your camera (supplied in the download package).
- Is the camera within the permissible range of the working distance?
- Does the working distance in your cell not deviate too much from the focus distance?
- Are your parts too small (see 9.1 Accuracy of the sensor)?



- Adjust lighting, eliminate strong / blending light sources, direct sunlight or reflections.
- Recalibrate camera (see camera instructions included in Quick Start Guide. [To download.](#))
- Adjust scan settings in camera SDK
  - More / longer acquisitions, use HDR feature
  - Vary parameters (aperture, gain, ...)
  - The goal is to get a point cloud as complete as possible (no gaps, 'black holes') without scattering, outliers, artifacts.
  - Make sure that the newly created configuration file is also used for the scan.

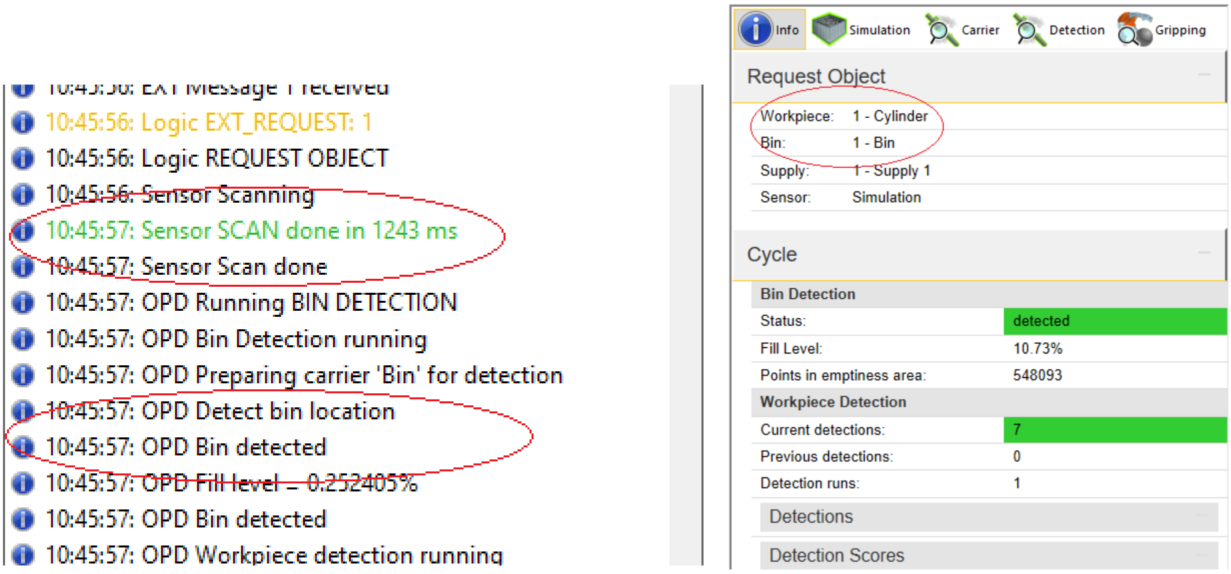


Refer to the operating instructions or the knowledge base of the camera manufacturer.

8.2 No workpieces were found – adjust detection parameters

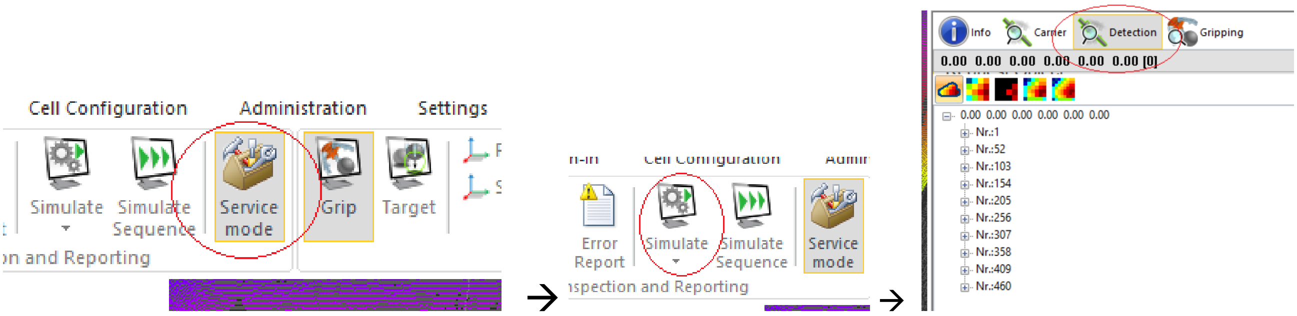
First of all, please make sure that:

- The scan was carried out successfully
- The correct box is searched for and found.
    - If the wrong box is searched for, change the settings under **Cell Configuration** > ID Table.
    - If the correct box is searched for but not found, check the stored CAD model (especially the models 'Complete' and especially 'Border', as the latter is used for the search) and your region of interest, see chapter 6.76.9.
- If the box is not found, proceed as described here, but in the 'Carrier' tab instead of 'Detection'.
- The search is performed for the right workpiece

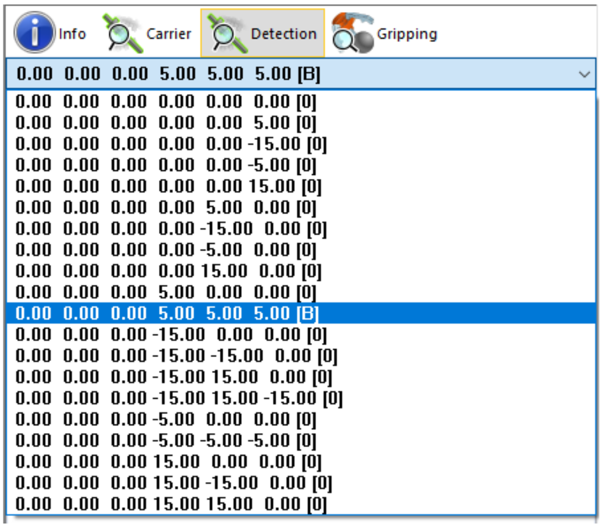


c.

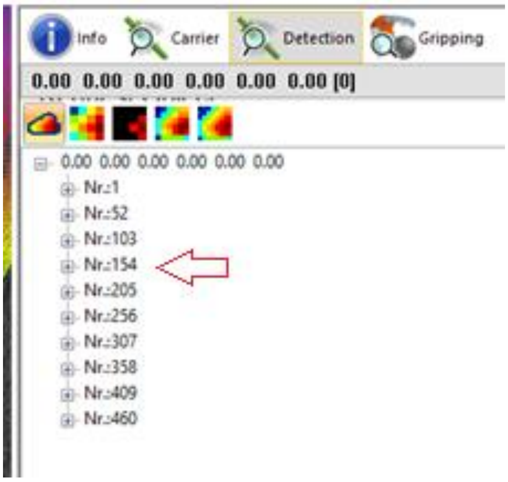
Checking the detection parameters:
To do this, please activate the service mode, scan again and open the detection tab.

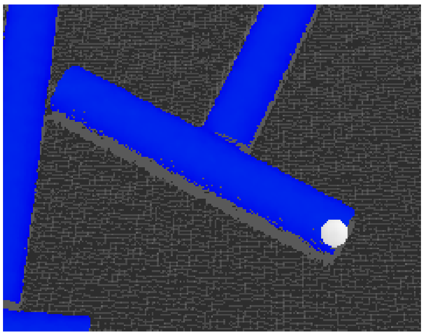


You should now see the following window.
Select the box at the top.



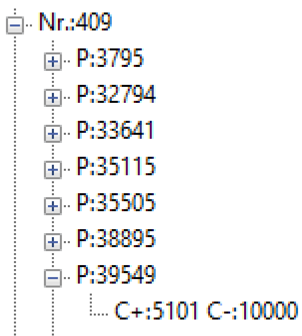
All possible views that the algorithm for the start point search displays are shown here.
Select the entry here with [B]. The best detection found is located here.
If no good detection is found here, look through the other possibilities as well.

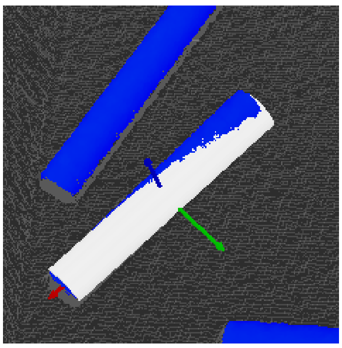




Each point listed below marks a starting point that is used for the workpiece search. If you click on one, it will be marked as a white point in the point cloud.

If it is the workpiece that is to be found, open the highlighted tab.





All positions the algorithm uses for the search are shown here.

View all positions and unfold the tab that best matches the position of the workpiece in the point cloud. The match doesn't have to be exact here, as it will be improved in the next step.

In the window below you can see now the reason why the workpiece was not recognized.

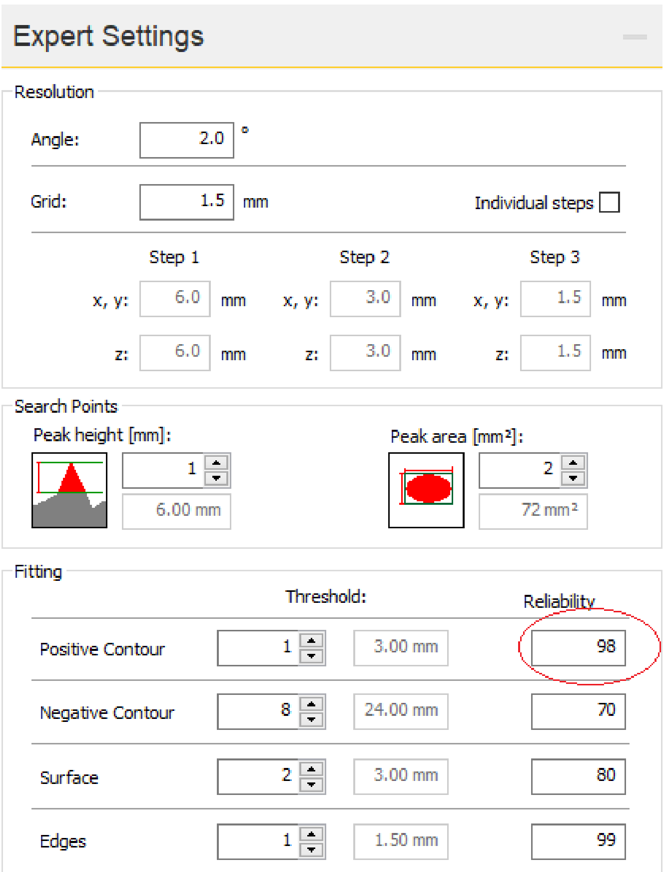
Property	Value
Step	OPDStep::CONTOURS
SearchDuration	1 ms
ScoreContourPositive	5101 (9800)
ScoreContourNegative	10000 (7000)
InvalidReason	P. Contour Score to low (5101)

The apparent reason given is: 'positive contour is too small'

Positive contour:	Value found: 51	Set minimum value: 91	→ not ok
Negative contour:	Value found: 100	Set minimum value: 70	→ ok

In this example the found value of the positive contour is smaller than the one defined in the detection settings.

Now switch to the detection settings and set the value of the positive contour to below 50.



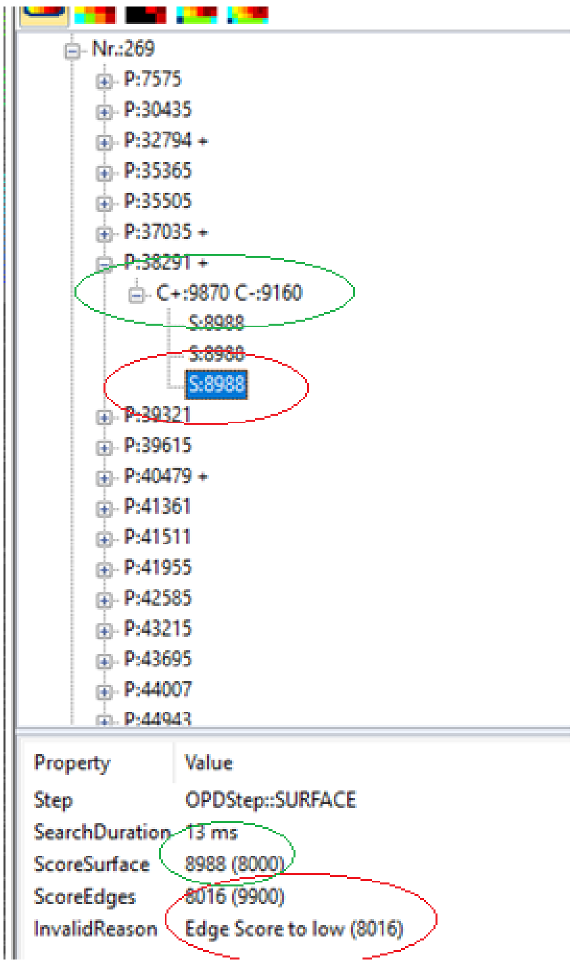
Now simulate again. If the workpiece is still not found, proceed again as described above.

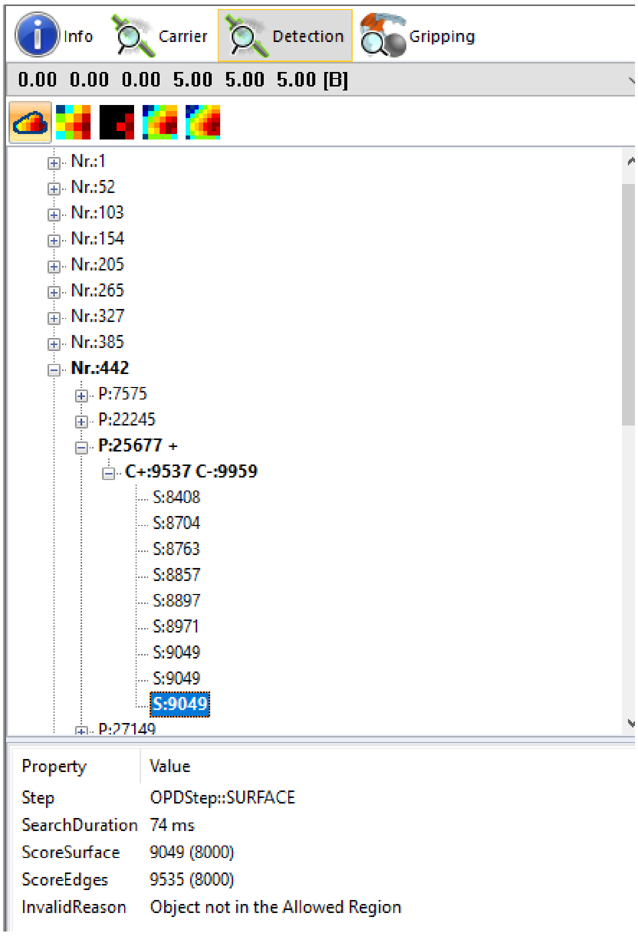
In this example you can see the following:

- The values for positive and negative contour are OK
- The value for surface is okay
- The value for edge, on the other hand, does not

To fix this, switch back to the detection settings and change the edge value accordingly.

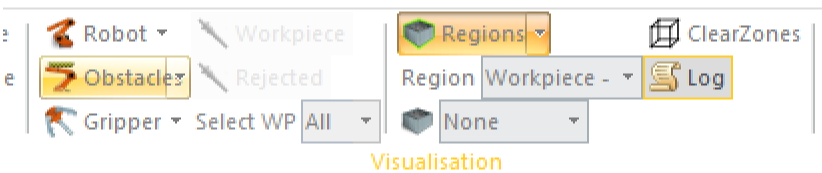
It can also happen that all detection settings fit, but there are invalid settings in the box.



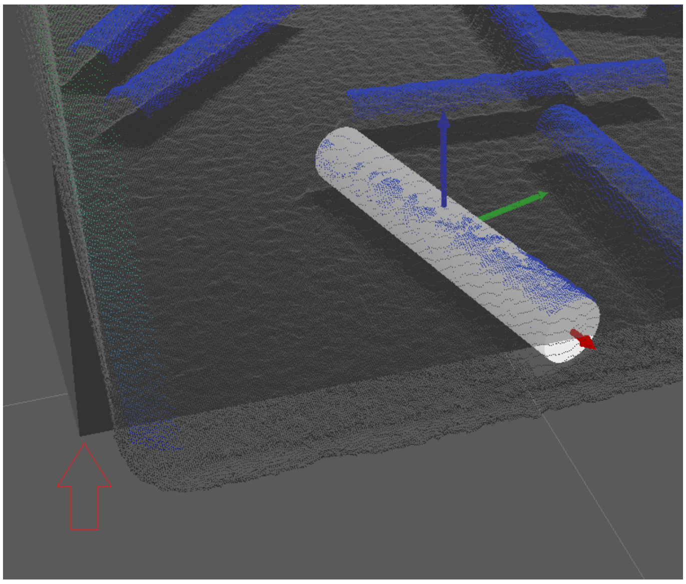


Here you can see, all values are ok, but the workpiece is not in the allowed region of the box.

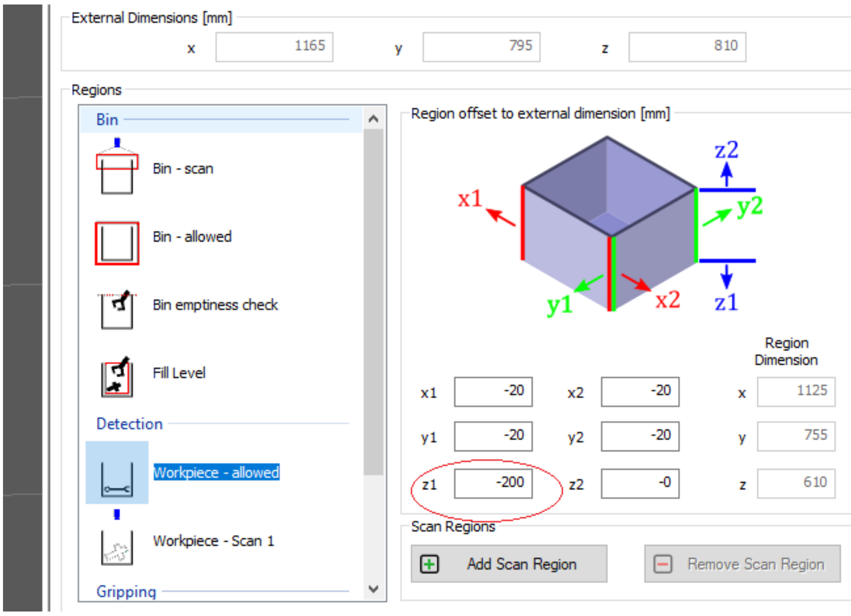
To check this, activate the regions in the 3D view and select 'Workpiece - allowed'.



You should now see the following in the 3D view:



Here you can see that the region in which the workpiece may lie is above the workpiece. You change this in the box settings.



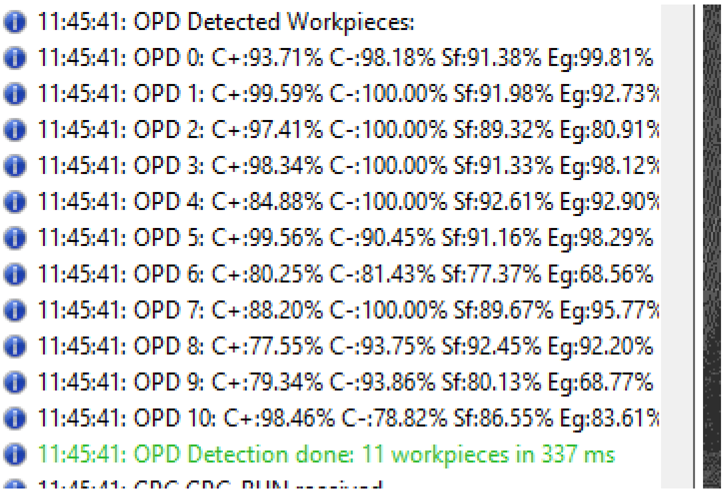
Now the workpiece should be found.

8.3 Incorrect workpieces are found

It may happen that false detections of workpieces occur if the detection settings are set too loosely.

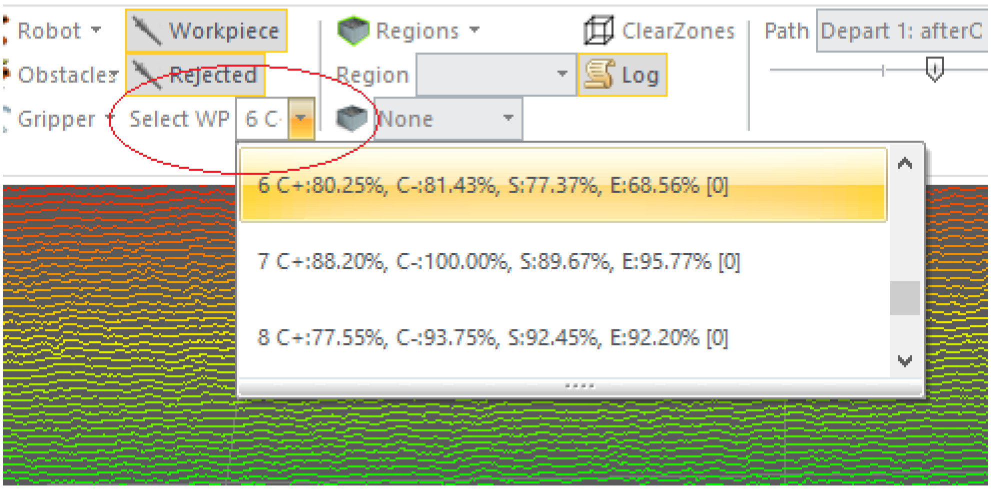
This usually happens when there are several different workpieces in a box.
It is recommended that only the workpieces to be searched for are in the box!

In the left list you can see all detected workpieces, the valid ones and also the invalid ones, each with a number at the front.



Now to find out which workpiece in the list is the invalid one, you can look at the individual workpieces.

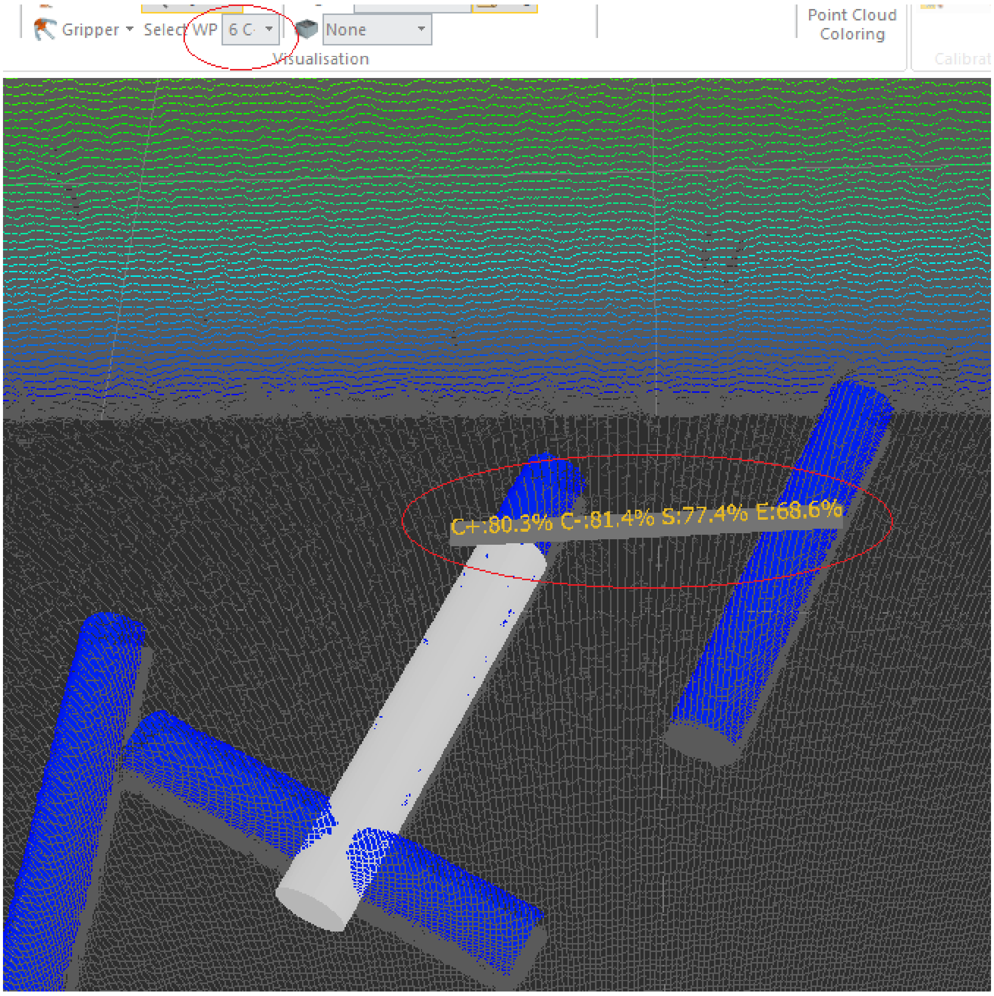
To do this, click the following in the 3D view:



Here you can select all workpieces individually.

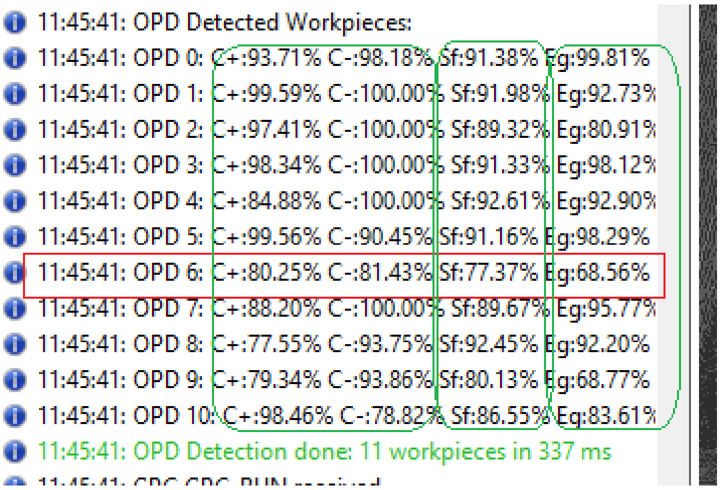
It is faster with the following key combination:
CTRL + Arrow Up / Down

Now look through all list entries until the workpiece that was incorrectly detected is shown.



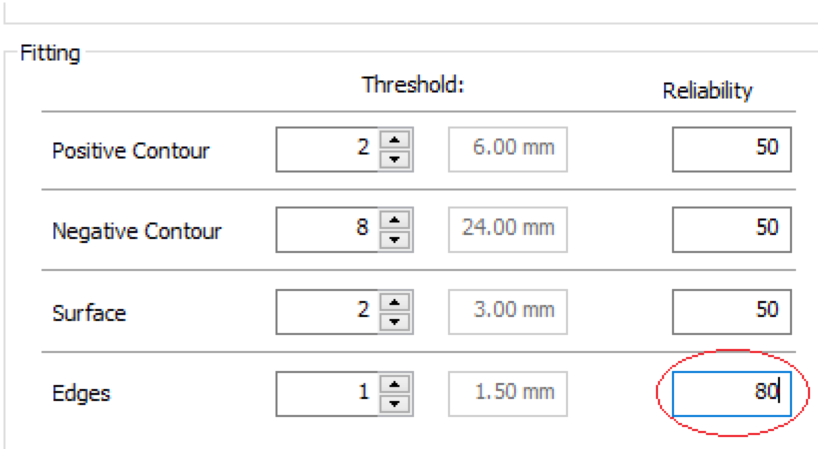
At the top, you will now see the number with which you can identify the workpiece in the left bar. Here it would be the workpiece number 6.

Alternatively, you can see the detection values as an overlay directly in the 3D view when the mouse is over the workpiece.



As you can see in the picture, there are 2 workpieces with worse edge values than the others. The selected workpiece 6 and also workpiece 9.

All others have values above 80, so it makes sense here to set the edge value to 80.
Thus, all workpieces with a value worse than 80 will not be detected.



If this is not enough, please proceed as follows:

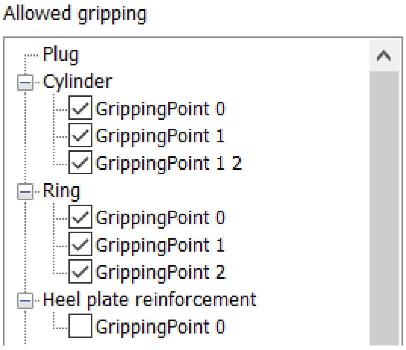
1. Increase edge value (as marked in the picture)
   If this does not help, increase the value at Edge Threshold by 1 to decrease the accuracy of the check
2. if this does not help, do the same with the Surface value
3. Only then adjust the positive and negative contour

8.4 No gripping point found

Result	
Result:	No Grip Found
Scan Time:	2.70 s
Calculation Time:	19.53 s
Total Time:	22.23 s

Note: The long calculation time indicates that many gripping points were calculated and is not an error.

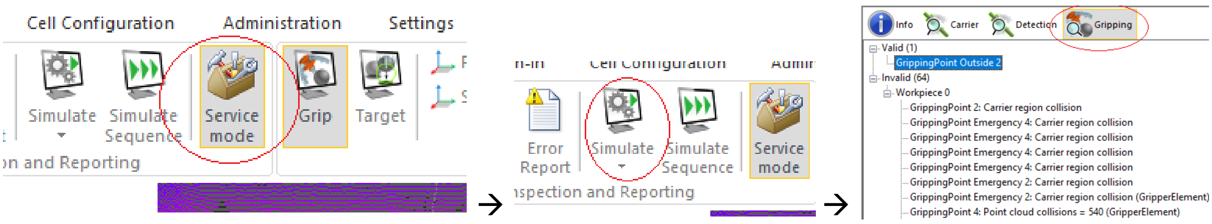
Check if all gripping points of the part are allowed for the used supply location under **Cell Configuration > Supplies > Allowed gripping**. That is the case if in the logging area (Operation > left status list) 'Gripping point not allowed for this supply' is displayed.
Only the gripping points that are checked are used:



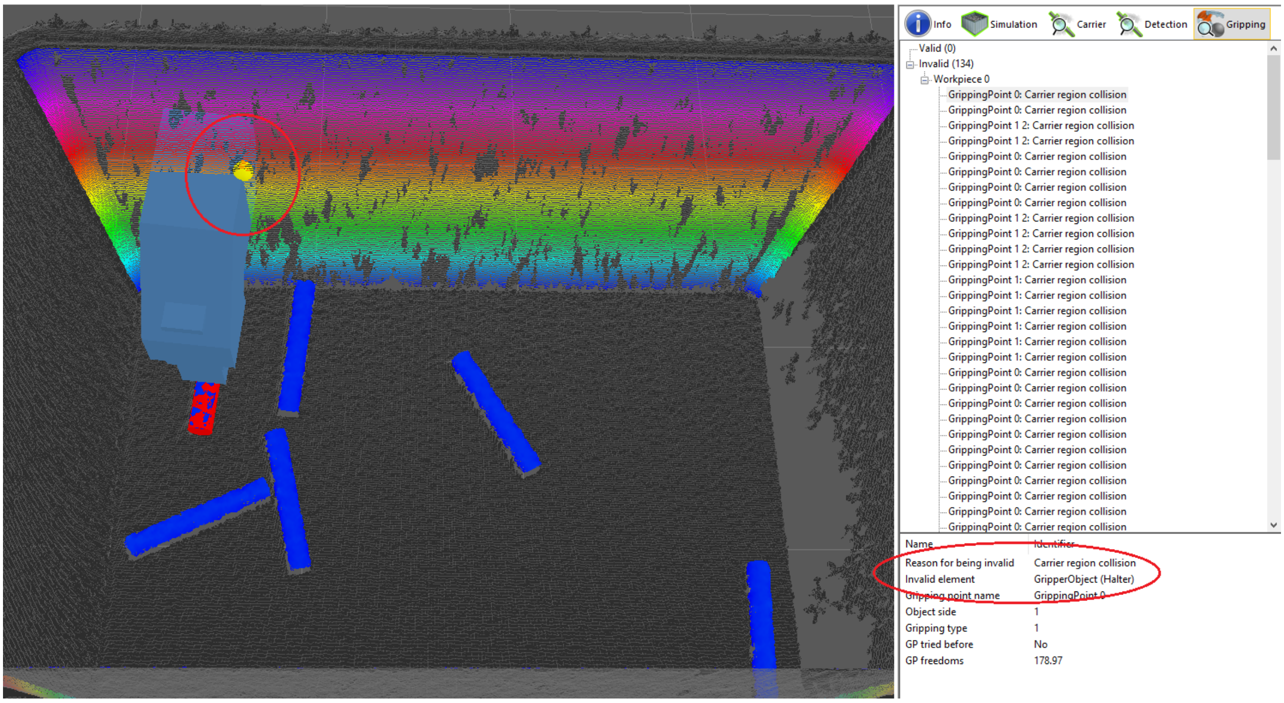
- Add new gripper points to allow more possibilities.
- Vary your gripper / the gripper model in the software

Checking the gripping points

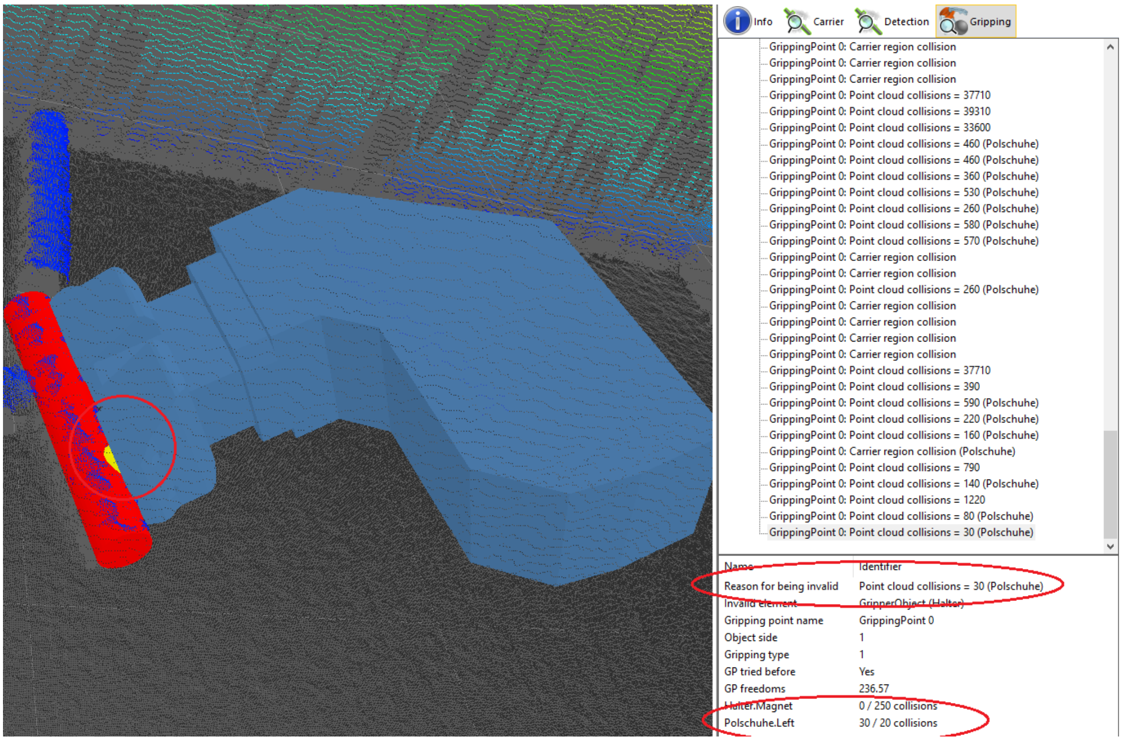
Activate the service mode, simulate again and open the 'Gripping' tab.



In the list all gripping points are shown that have been calculated by the software.
As soon as you select a gripping point in this list, it will be visualized in the 3D scene and you will see why this gripping point was not found.



In this example, you can see that the gripper would collide with the box. This is displayed in the information window at the bottom right.



In this example, there is a collision between the gripper and the point cloud (yellow point). The information window shows which element of the gripper collides with the point cloud. You have the following options to fix this:

- The gripping point is too close to the part and must be set with some distance.
- Increase the number of valid collisions with the gripper (especially like here with magnetic grippers, where component and gripper touch each other).

Due to the fact that the point cloud always has some ‘noise’, i.e. points in space that do not represent actual objects in space, collisions will always occur. However, if this only occurs at individual points in the point cloud, this is not an obstacle.

9. Troubleshooting: Accuracy

Having problems with the accuracy of your bin picking application? Our guide will help you find the solution.

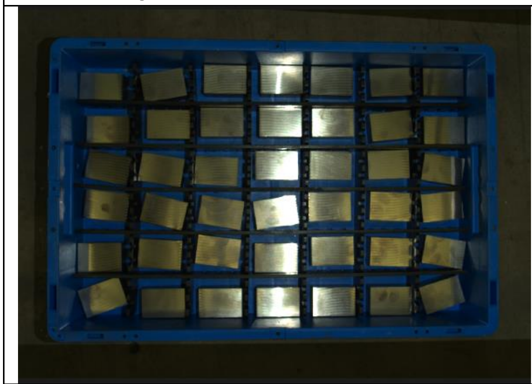
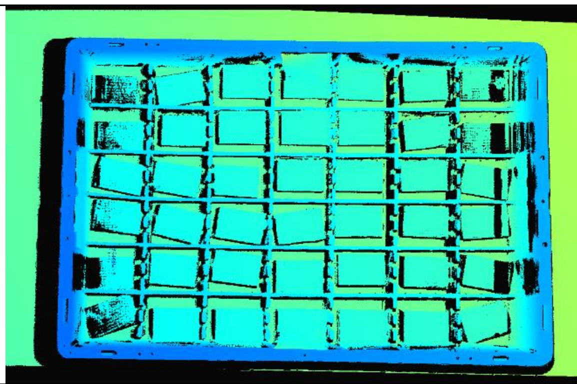
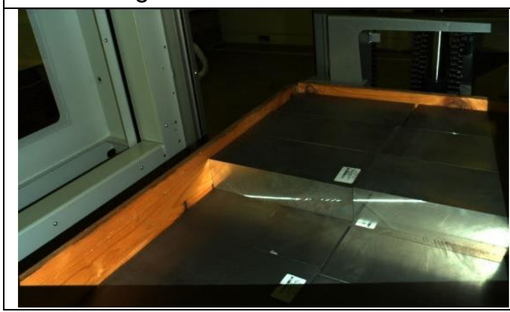
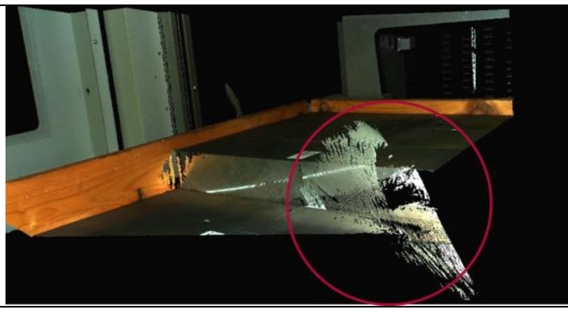
9.1 Accuracy of the sensor

9.1.1 How do I recognize the problem?

- Parts are not detected.

Result	
Result:	No Object Found
Scan Time:	6.93 s
Calculation Time:	5.39 s
Total Time:	12.32 s

- Point clouds are patchy or have strong outliers (so-called artifacts).

<table>
<tr><td colspan="2">Example 1: patchy point cloud
Black areas = ‘blind spots’, no points found in this area.</td></tr>
<tr><td>Real image</td><td>Point cloud</td></tr>
<tr><td></td><td></td></tr>
<tr><td colspan="2">Example 2: Artifacts
Points are found where in reality there are no objects. Occurs especially with reflections and mirroring.</td></tr>
<tr><td>Real image</td><td>Point cloud</td></tr>
<tr><td></td><td></td></tr>
</table>

9.1.2 How can I check it?

View the point cloud and check for the following criteria:

- Clean and clear edges.
- Are small features, e.g. slits, gaps well detected?
- Continuity of the point cloud: are there holes? Is the point cloud complete? Is the point cloud flat or wavy?
- Zivid: increase dimensional accuracy using [InfieldCorrection]

- Check pixel resolution: Is the resolution of the sensor sufficient for detecting small features?
  - Field of view in x-direction / long side (FOV).
  - Sensor resolution = number of pixels in p_x and p_y (e.g. 1920x1200)
    - Zivid One+ with 2.3MP = 1920 x 1200
    - Zivid Two with 2.3 MP = 1944 x 1200
    - Ensenso X36 with 5 MP = 2448 x 2048
    - Ensenso N35 with 1.3MP = 1280 x 1024
    - e.g. field of view 500 x 300mm with Zivid One+

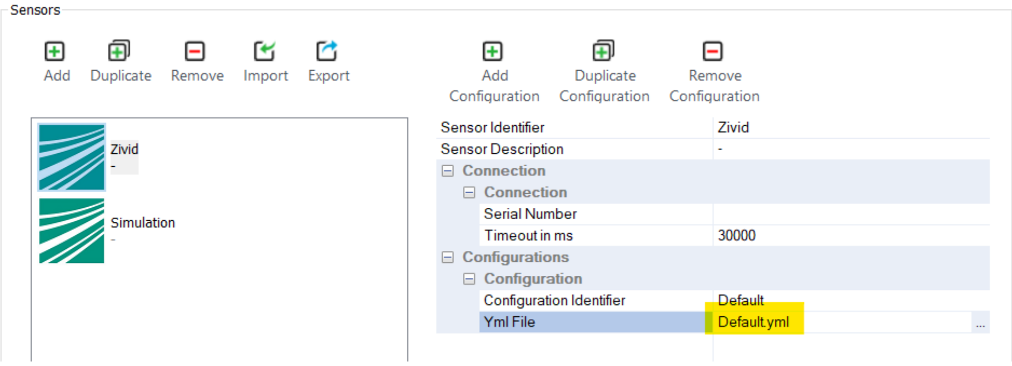
$$Pixel\ resolution = 500mm/1920\ \ pixels = 0{,}26mm/pixel$$

    - Pixel resolution in y-direction is identical; adjusted via camera.
  - What is the size of the smallest feature? Can it be detected with the existing resolution?
- Is the camera technology the right one for your application?

Procedure	Pro	Contra	Manufacturer
Laser Triangulation	– Inexpensive, simple and proven method – Good accuracy – Good for detection of moving objects on conveyors	– Difficulty with shiny and dark surfaces – Slow with stationary objects (camera must move on axis)	Automation Technology (z.B. C6) Wenglor (z.B. Shapedrive)
Stereo Vision	– Good detection of edges – Well suited for long distances and variable ambient light	– Shadow casting – Long scan and calculation time for round objects – Deficent detection of planar surfaces	Nerian (z.B. Scarlet)
Active Stereo Vision	– Like Stereo Vision, but faster calculation and better recognition of surfaces	– Like Stereo Vision, but more expensive	Ensenso (z.B. X36)
Structured light	– Complete 3D image of the object, good accuracy – Good recognition of round shapes – Good recognition of planar surfaces due to pattern projection – Short calculation time, less processor-intensive	– Ambient light can impair result – Working distance limited, as less and less light arrives with increasing distance – Difficulties with shiny and dark surfaces	Photoneo (z.B. PhoXi 3D)
Time coded structured light	– Like structured light – All calculations on pixel level: block noise and loss of spatial resolution is avoided – High accuracy	– Like structured light – Longer calculation time – Camera and object must be stationary	Zivid (z.B. One+)
Time of Flight (also: LIDAR)	– At working distances < 10m less accurate than structured light – At working distances > 10m more accurate than structured Light – Very fast method – All calculations on pixel level: block noise is avoided	– Low spatial resolution	Ifm (z.B. O3D) Sick (z.B. Visio-nary-T Mini) Basler (z.B. Blaze)

A deficient point cloud can …
- result from bad environmental conditions → see 9.1.3 How do I fix it?
- occur with challenging e.g. reflecting / mirroring parts → see 9.2 Detection accuracy

9.1.3 How do I fix it?
- Check the working and focus distance using the data sheet of your camera (supplied in the download package) and adjust if necessary. If you change the working distance, the hand-eye calibration must be performed again; in case of grave changes, it is also recommended to recalibrate the camera.
- Recalibrate camera (see camera instructions included in Quick Start Guide. To download.)
- Adjust lighting, eliminate strong / blending light sources, direct sunlight or reflections.
- Adjust scan settings in camera SDK
  - More / longer acquisitions, use HDR feature
  - Vary parameters (aperture, gain, …)
  - The goal is to get a point cloud as complete as possible (no gaps, 'black holes') without scattering, outliers, artifacts.
  - Make sure that the newly created configuration file is used for the scan.



Refer to the operating instructions or the knowledge base of the camera manufacturer.

Goal: Point cloud as complete as possible so that workpiece is detected

9.2 Detection accuracy

9.2.1 How do I recognize the problem?
- Point cloud (see 9.1 Accuracy of the sensor) is good, no gaps, no artifacts, but part is not detected
- Error message 'No object found'

Result	
Result:	No Object Found
Scan Time:	6.93 s
Calculation Time:	5.39 s
Total Time:	12.32 s

9.2.2 How can I check it?
- Is the correct part being searched for? Check ID Table and query of the robot (robot program, settings under **Operation** > Simulate.
- Check CAD model, does this match the actual object?
- Check orientation e.g. if stacked workpieces are expected and workpieces are in chaotic storage, they are not recognized.
- Is none of the parts completely visible? Then search for segments.
- Are the features of the object recognizable by the sensor? Check accuracy values see 9.1.2

9.2.3 How do I fix it?

- Correct ID table
- Adapt CAD model
- Adjust orientation according to actual conditions under **Teach-In** > **Detection** > Orientation Settings
- Segment search – instructions for setting up see 6.2.3
- In the worst case the camera is not accurate enough for your part and has to be replaced.

9.3 Hand-eye calibration or TCP

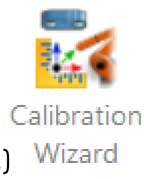
9.3.1 How do I recognize the problem? Where does it come from?

- Systematic error e.g. robot gripper always 5mm away from target gripping point.
- Random error – robot gripper is sometimes positioned correctly (at the target gripping point), sometimes off, but no pattern is visible. This indicates incorrect Hand-eye calibration.
- Incorrect Hand-eye calibration = camera and robot are not in the stored position. This can occur if the position of an object was changed after calibration, e.g. the camera was mounted differently

9.3.2 How can I check it?

- Physically mark the gripping point on the actual workpiece and perform 10 picks. The workpiece should always be gripped at the same position. Observe whether gripping always (systematically) occurs e.g. too far to the left, or whether an angular error occurs (e.g. if object is placed down at a slight angle with each pick).
- Alternatively, check with a measuring tip. To do this, create a gripper with a measuring tip in LHRobotics.Vision, move it to a defined gripping point and check whether the robot moves to the correct point based on the scan.
- Does the gripper model in LHRobotics.Vision match the correct gripper? Is the position of the gripper displayed correctly?

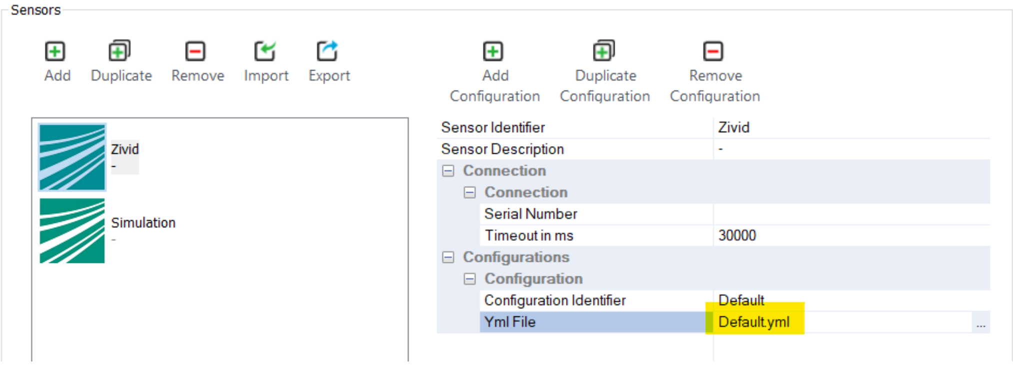
9.3.3 How do I fix it?



- Correct gripper model
- Re-teach TCP
- Check robot program
- Retry hand-eye calibration see chapter 5 Hand-eye-calibration.
  - Is the plate detected in all cases? (reflections due to strong light sources)
  - Does the robot collide with anything during calibration?
  - Does the software report ‘successful’ at the end?

9.3.4 Possible problems during hand-eye calibration

- Calibration plate is not detected – unsuitable exposure settings
  - Open camera SDK
  - Adjust the acquisition so that it fits the scene. See camera manufacturer's documentation for tips.
  - Import adjusted .json/.yml file in sensor settings.
  - Recalibrate



- When only some poses are not recognized: Outliers are automatically calculated out by the program. This does not represent an error.

If the calibration is completed successfully but the error still persists, contact Liebherr Service, in case the calibration process is not suitable for your camera.

9.4 Accuracy of the robot / kinematics

9.4.1 How do I recognize the problem?

- Path planning not possible or not correct
- Gripping points and/or placement of the workpiece is inaccurate

Note: Accuracies < 2mm are challenging and often require an additional camera to check the exact location.

9.4.2 How can I check it?

- Is the robot kinematics 6-axis? LHRobotics.Vision Pro, version 3.2 can only perform collision checking and complete path planning for robots with 6-axis kinematics.
  - No 5- and 7-axis robots, SCARA or similar possible.
  - Only relevant for Pro version. Robot kinematics are not considered in the Basic version.
- Check requirements for robots
  - See 1.2 Requirements regarding the robot
  - For questions on the requirements, contact your robot supplier.