optonic
vision · automation

Updates

mikado

Version 4.0

A product of Optonic
www.mikado-robotics.de

Optonic GmbH | Zollhallenstr. 11 | 79106 Freiburg | Deutschland/Germany
Tel.: +49 761-79095-0 | E-Mail: info@optonic.com
© Copyright Optonic GmbH, 2015-2026

Contents

1 Mikado **2**
- 1.1 Version 3.5.0 . . . 2
- 1.2 Version 3.4.0 . . . 7
- 1.3 Version 3.3.0 . . . 9
- 1.4 Version 3.2.0 . . . 16
- 1.5 Version 3.1.2 . . . 18
- 1.6 Version 3.0.5 . . . 22
- 1.7 Version 3.0.4 . . . 22
- 1.8 Version 3.0.3 . . . 22
- 1.9 Version 3.0.2 . . . 23
- 1.10 Version 3.0.1 . . . 23
- 1.11 Version 3.0.0 . . . 23
- 1.12 Version 2.1.0 . . . 28

2 Scene objects **28**
- 2.1 Version 5 . . . 28
- 2.2 Version 4 . . . 30
- 2.3 Version 3 . . . 30
- 2.4 Version 2 . . . 30

This chapter describes changes in Mikado versions in detail. A complete and always up to date of all components of the Mikado system can be found on our website at https://optonic.com/support.

1 Mikado

Below you can find all changes for all Mikado versions.

1.1 Version 3.5.0

Mikado 3.5 extends motion planning for removing parts from a box. New timing and error-report features help improve program sequences, while further changes make cells easier to set up and edit.

Plan retract in sections

In Mikado Control, the `In Sections` interpolation type divides the retraction after extraction into individually configured movements. For example, the held part can first be aligned parallel to the box walls and then moved straight upward. Sections are planned in the configured order.

`Align with Box` provides target values and tolerances for the part's position and orientation relative to the box. `Drive Straight Upwards` moves the part along the positive Z-axis of the box coordinate system to the selected target Z value. Below the section list, you can choose whether planning continues to the transit pose.

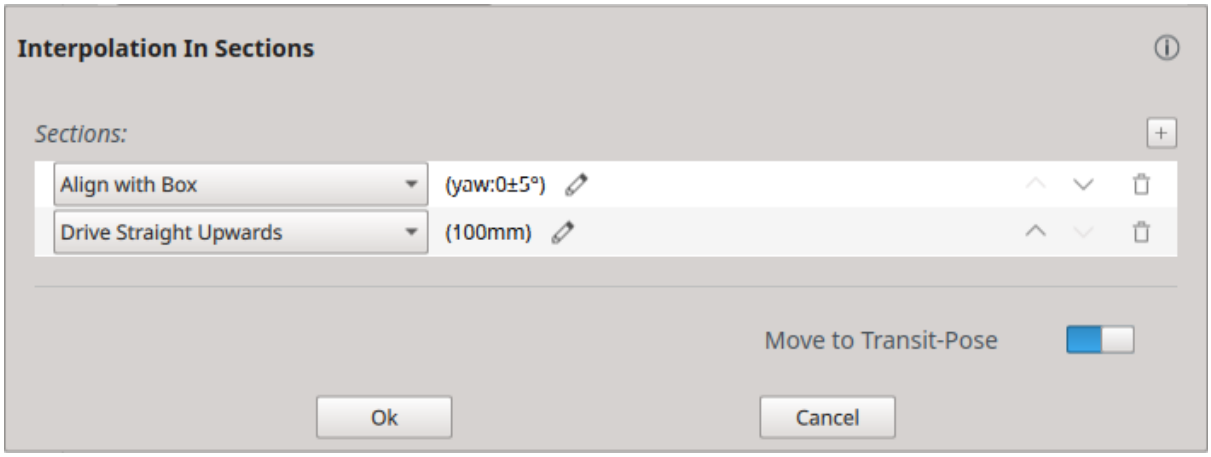

Fig. 1: The section list defines the order; continuing to the transit pose is a separate option.

The individual settings are described under Interpolation in Sections.

Control allowed collisions during extraction

The search strategy now distinguishes collisions of the held part with other recognized objects, the point cloud (Octomap), and the box. For each group, you can independently choose which contacts are allowed and set their maximum collision depth.

`None` disallows collisions in that group. `GraspPoint` only allows contacts already present at the grasp point, until they first disappear. It does not allow the same contact to occur again later. `All` also allows new contacts within the configured depth. This can be used, for example, to tolerate small overlaps caused by object recognition.

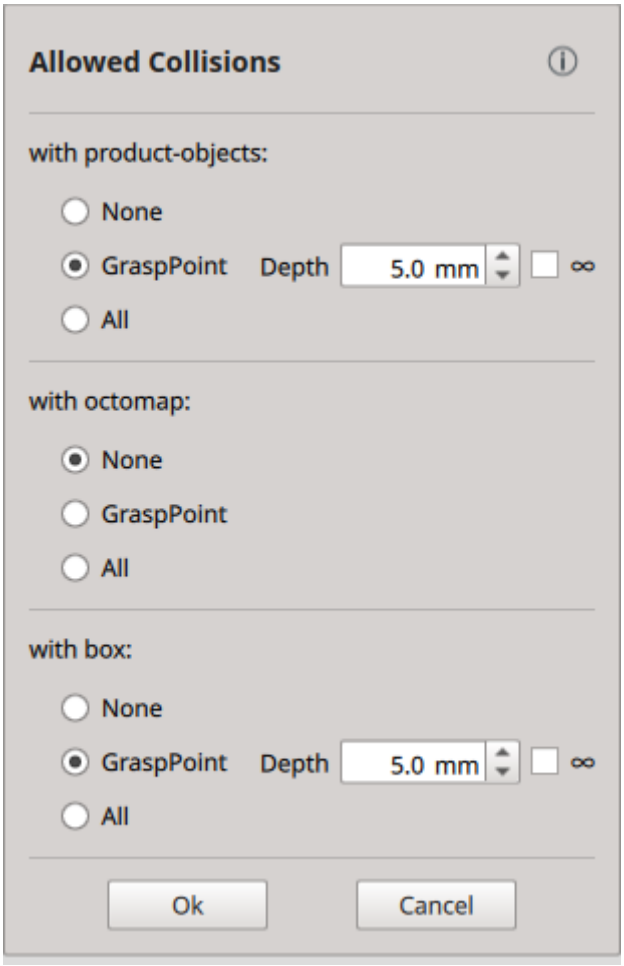

Fig. 2: Each combination defines the allowed collisions for all three groups.

You can add several combinations as alternatives. The pick search tests each extraction function with each combination. Pick inspection provides more detailed collision information and explains unmet extraction conditions. For details, see *Entnahme*.

Measure cycle times with the Stopwatch task

The new *Stopwatch* task measures the duration of a sequence directly in the Mikado program. Use `Start / Split` before the sequence and `Stop` afterward. Both tasks must use the same `Label` value, for example `PickCycle`.

The optional `Interval [s]` output contains the measured time in seconds. For reoccurring sequences, you can instead call `Start / Split` at the same point in each cycle: The first call returns `0`, and all subsequent calls measure the time elapsed since the previous call. `Median [s]` provides the median of the last up to 1000 recorded intervals for that label.



Fig. 3: Using the same label connects start, split measurements, and stop to one timer.

Choose what to include in error reports

When creating an error report on the system page, you can choose whether to include the last camera capture as a filecam and a complete cell backup. Add a note describing the observed problem and when it occurred. The filecam helps you examine the last captured scene again later.

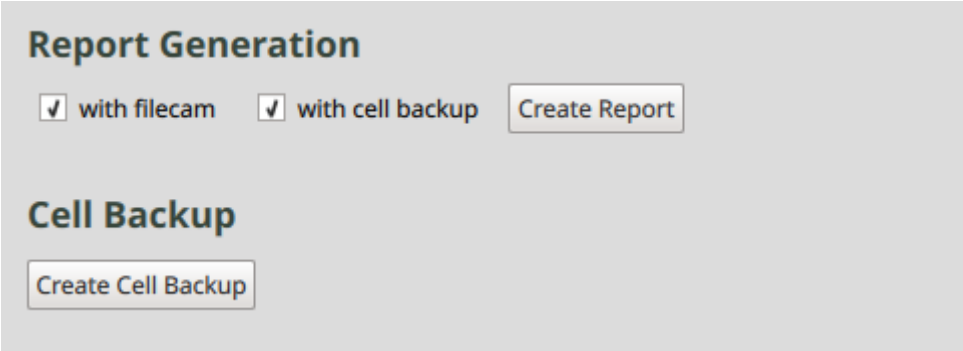

Fig. 4: The additional report contents can be selected independently.

The new *Save Error Report* task lets a program create a report itself, for example in an error-handling sequence. It can optionally include the last point cloud and a note explaining the situation; the task comment is included as well. Creating an error report takes time and is therefore only suitable for specific diagnostic cases.

Cell backups are now saved as `tar.gz` archives with the cell name and a timestamp. If `pigz` is available, compression uses multiple processor cores. The standalone logger can also directly open log files and entire log folders passed to it.

Tolerance parameters for 2D box search

The 2D box search can evaluate invalid results by specifying a maximum permitted deviation in position and angle, see *Kistensuche*. If either of these limits is exceeded, the search is considered to have failed.

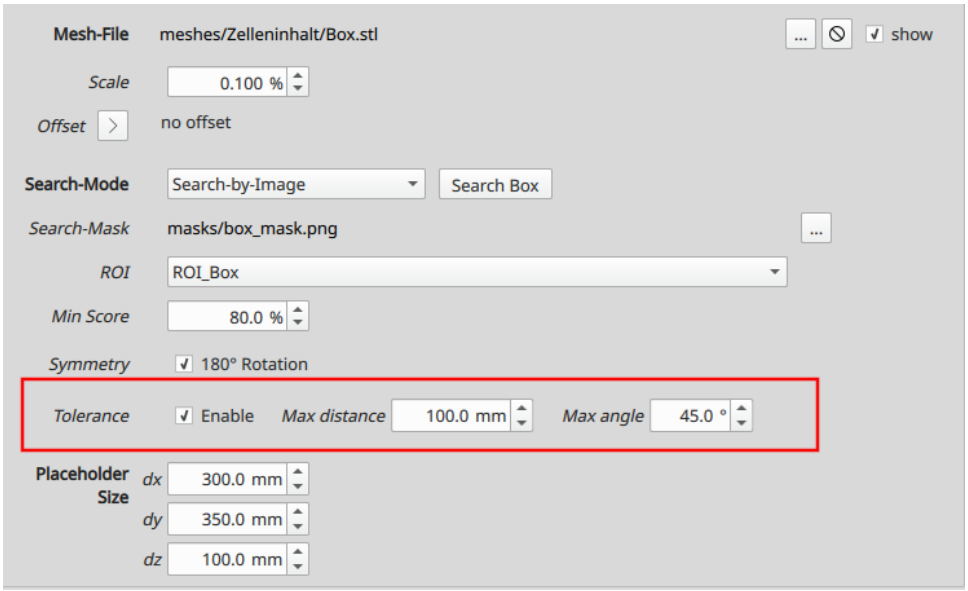

Fig. 5: Tolerance parameters for 2D box search

Generating product geometry from parametric blueprints

Product geometries can optionally be generated from predefined parametric blueprints, rather than specifying a mesh file. The geometric shapes available for this purpose are cuboids, spheres, cylinders and hollow cylinders. The parameters of the blueprints can be modified both via the user interface and via JavaScript during program execution.

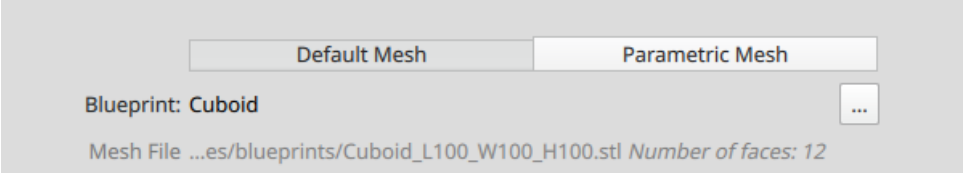

Fig. 6: Selecting the parametric blueprint in the product settings

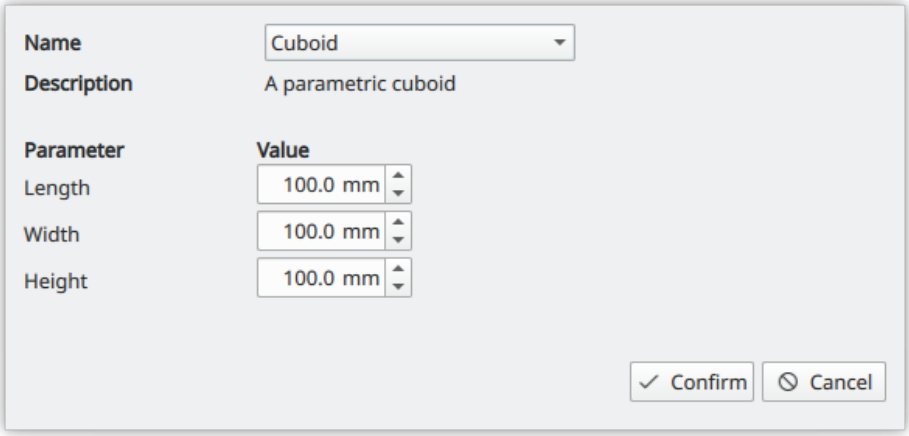

Fig. 7: Dialogue for editing a parametric blueprint

Further improvements

- **Programs:** New standard sequences for Mikado Control and Select, plus an example for hand-eye calibration using fixed poses, have been added. These can be found in the program library on the Welcome page. Loaded example programs now also open directly in the *Program editor*.
- **Loading files:** When opening external program, product, scene or mesh files, these can be copied automatically into the cell.
- **Calibration:** Before automatic hand-eye calibration, the active gripper and the appropriate scene must be selected in order to avoid collisions caused by carelessly chosen settings. The movements limit the speed and acceleration scaling to a maximum of 10%. No free path planning is used.
- **Picks and Pick inspection:** The selected search identifies its camera, ROI, parameter sets, and search strategies. *Handle inspection* also shows the ROI used when a search fails. Information on allowed collisions, unused trajectory steps and reasons for unmet picking criteria is now easier to understand. The tasks *Get Picks*, *Pick object* and *Pick-Search-Statistics* provide additional information on collision depth and the robot's joint position at the grasp point via optional output variables.
- **Variables:** In the *Variable editor*, names containing numbers are sorted in natural order, such as `Pose2` before `Pose10`. Use *Ctrl/Shift* to select multiple variables and *Delete* to delete them all at once. Units and localised number formats make task values easier to read. Pose variables can be displayed using other rotation conventions.
- **I/O and JavaScript:** The I/O page indicates whether signals are routed through the robot or fieldbus. JavaScript I/O calls accept an explicit namespace. New functions provide grasp names for a product and the current pose of the held part.

- **Terminology:** The term `experimental grasp` is replaced by `auxiliary grasp`. The position previously also referred to as the transition pose is consistently called the `transit pose`.
- **Display:** The Mikado application is displayed in full on screen resolutions of 1280×1024 and above. Views can be undocked and moved as grouped tabs (replacing the previous "separate windows" setting).

1.2 Version 3.4.0

Mikado version 3.4.0 extends the pick search, adds new programming options, and improves usability when starting, backing up, and setting up programs.

Extended Pick Search

The *Get Picks*, *Pick object*, and *Pick-Search-Statistics* tasks provide additional output variables. This allows programs to directly access information such as the number of recognized objects, the number of pickable objects, object pose, pick, gripper, and collision type.

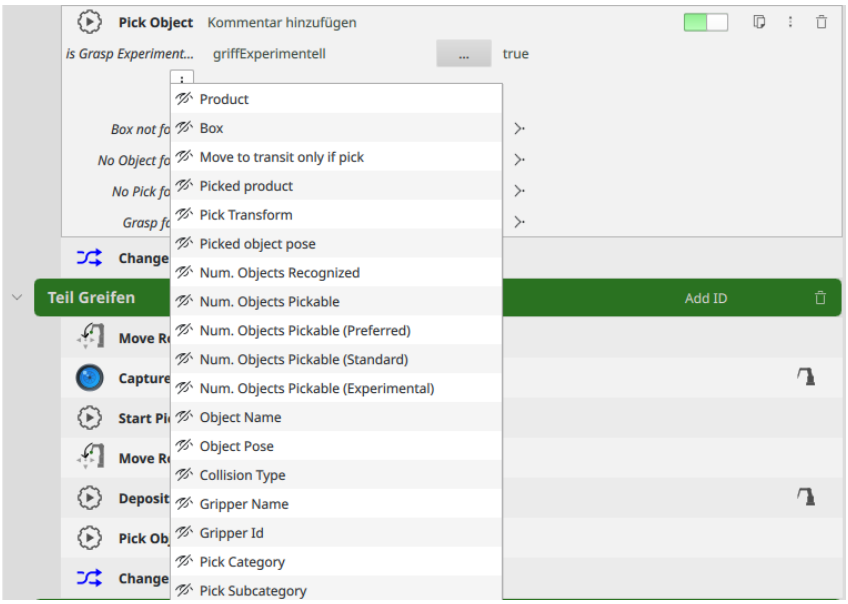

Fig. 8: Additional output variables in the `Pick Object` task.

A fixed Z height relative to the box can now be specified for the pick search. The Cartesian movements for picking and extraction are planned up to this height, making path planning easier to control for rods, tight boxes, or higher edges.

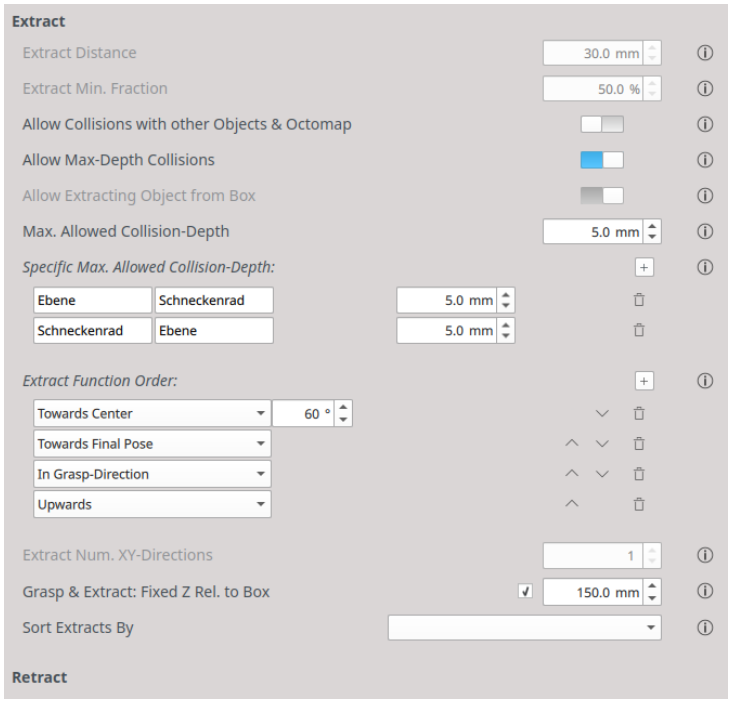

Fig. 9: Setting a fixed Z height relative to the box for picking and extraction paths.

Pick trajectories can also be smoothed. This creates smoother transitions between approach, pick, extraction, and retract.

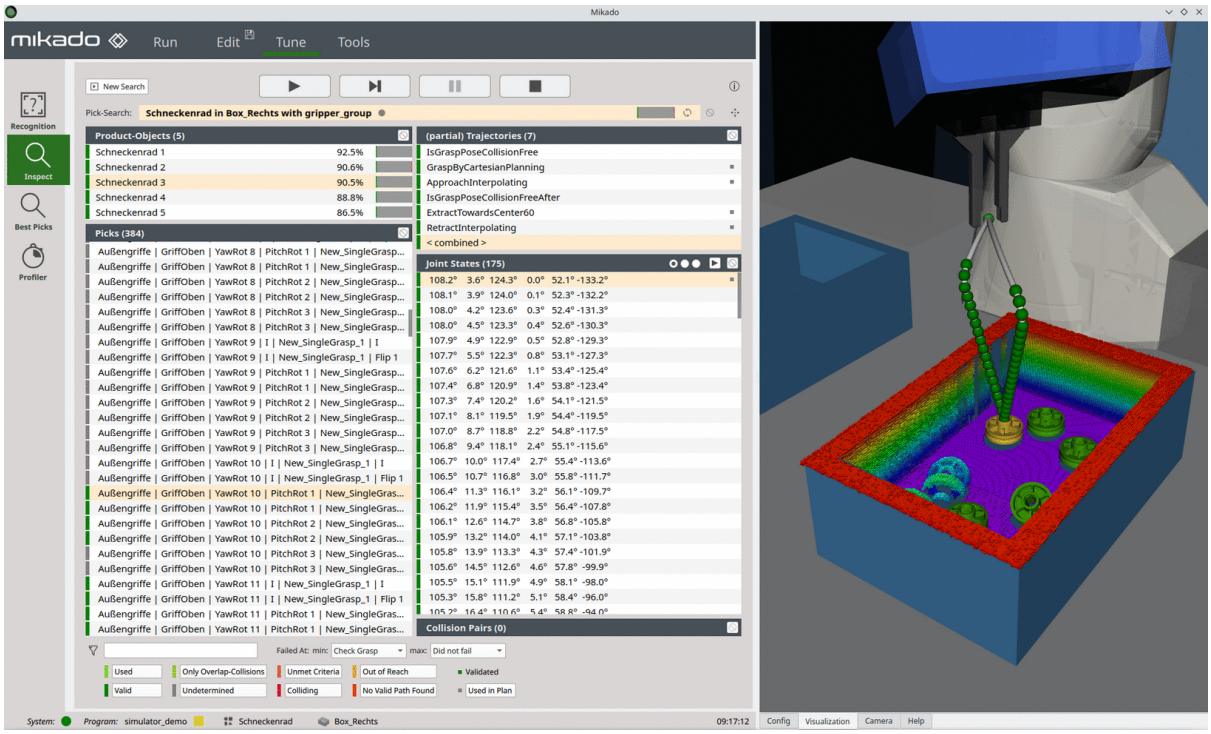

Fig. 10: Smoothed pick trajectory in the pick inspection.

New Tasks and Interfaces

- **Robot movements:** New tasks enable relative movements of the robot, TCP, or picked part.
- **Robot signals:** The new *Wait for robot output signal* task waits for robot output signals.
- **JavaScript:** The pose of the picked object can be queried via picksearch.getPickedObjectPose().
- **FANUC:** The FANUC-specific page can show the robot web interface and open it in the browser.

Improved Usability

- **Welcome page:** Startup files for programs, products, and scene objects can be selected directly.
- **System page:** The current cell can be backed up as a ZIP archive.
- **Program settings:** Language selection, window size, window state, and full-screen mode are saved on the respective controller after changes in Mikado. This makes it easy to use your own language during offline commissioning or troubleshooting, for example, without having to change the customer's cell.
- **Task Editor:** Optional variables without a value are initially hidden and can be added through a menu when needed.

1.3 Version 3.3.0

Mikado version 3.3.0 introduces various new tasks, a new fieldbus page, new error reporting functionalities, and includes numerous usability and stability improvements.

Greater flexibility for vacuum grippers

Starting with Mikado 3.3.0, an area can be defined for vacuum grippers that is not considered a collision object during grasp planning. As shown in the example below, a plane is defined relative to the TCP (with a definable Z offset). Among other things, this functionality allows flexible suction cups of vacuum grippers to be excluded from collision checks during grasp planning. The new function can be set on the gripper page of the Mikado Assistant.

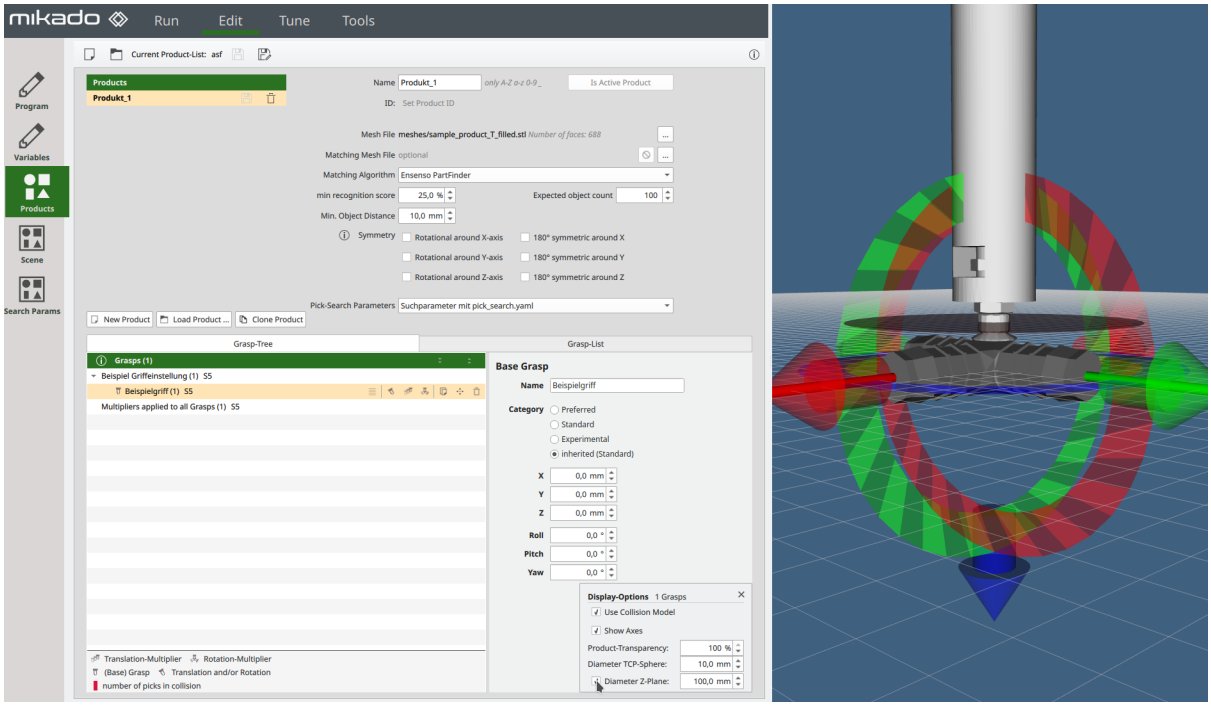

Fig. 11: Visualization of the area on the gripper that is not checked for collisions during the pick search. Everything below the dark gray plane is not treated as a collision object during the pick search.

In addition, for different robot manufacturers there now is the option of not activating the vacuum gripper at the transit position, but rather when a specified distance is reached during movement. This minimizes unwanted suction or displacement of parts before the actual gripping position is reached.

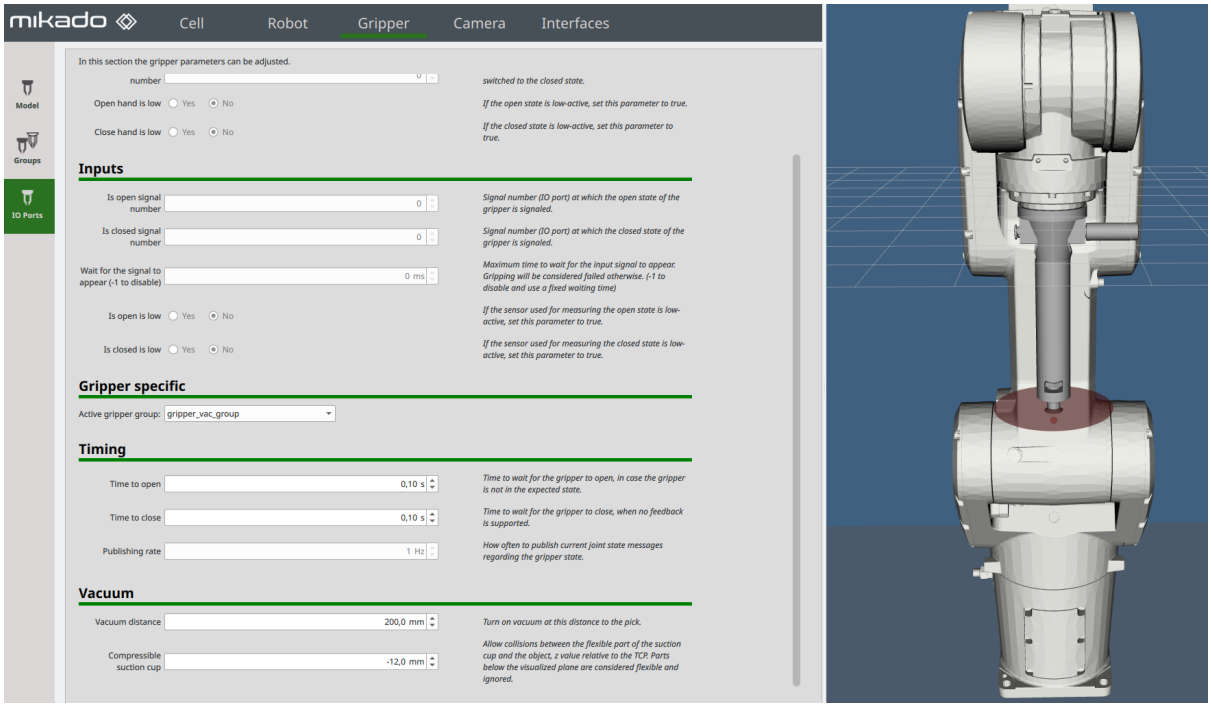

Fig. 12: Setting options for distance-based activation of a vacuum gripper and compression of the suction cup in the Mikado Assistant.

Additional filter options

To make complex programs easier to use, filter options have been added in various places. Variables and sequences can now be found more easily by entering their names in the corresponding text fields.

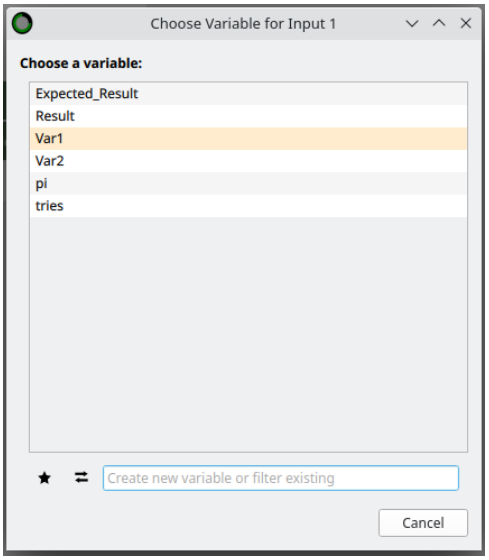

Fig. 13: New filter in the variable selection dialog

New filter options are available in the following locations:

- **Variable selection:** Filtering of boxes, ROIs, products, and other variables

- **Variable Editor:** Filter by Type
- **Sequence selection:** Filter by name
- **Program Editor:** Filtering when jumping to sequences
- *Best picks:** New filter to display only picks that were available at the time of pick selection
- **Pick inspector:** Filter by pick name

New and revised tasks

Three new calculation tasks are available in version 3.3.0: *Arithmetic operation*, *Logical operation* and *Comparison operation*. The *Arithmetic operation* task allows simple arithmetic operations to be performed with two variables (integer or double), the *Logical operation* task can be used to implement boolean operations, and *Comparison operation* allows two variables to be compared.

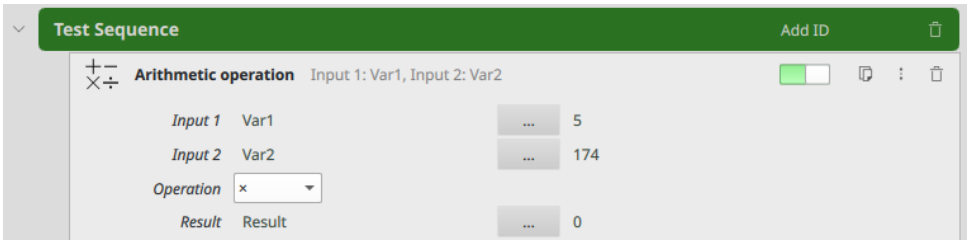

Fig. 14: New *Arithmetic operation* task

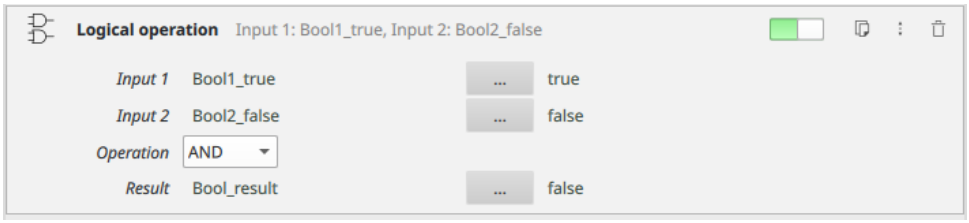

Fig. 15: New *Logical operation* task

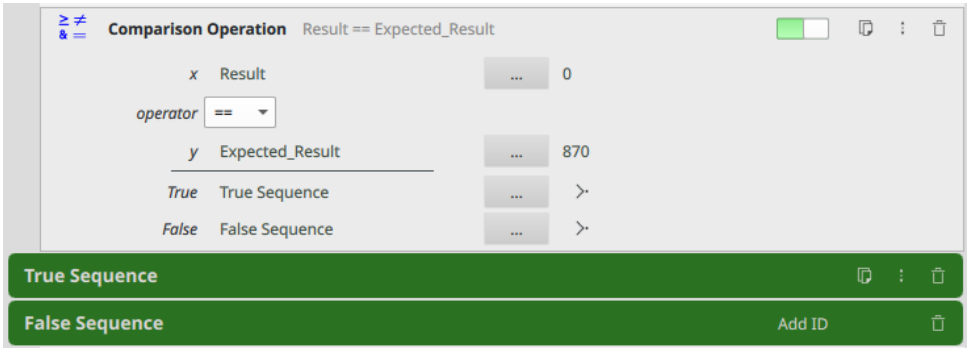

Fig. 16: New *Comparison operation* task

Another new feature is the *Set Fieldbus Output Signal Async* task, which sets a fieldbus output as soon as the robot has left a specified ROI. This enables convenient and fast feedback to the PLC when the robot has left a specific area.

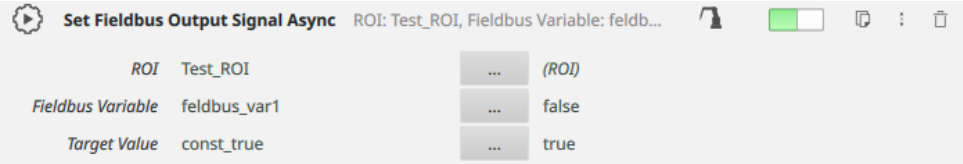

Fig. 17: The new *Set Fieldbus Output Signal Async* task

Further changes to tasks:

- *Create pose* can now also be used to overwrite a single component (e.g., the Z value)
- *Find object*, *Start handle search* and *Get Picks* no longer make changes to the active product
- In the *Pick object* task, an optional parameter can be set to ensure that the move to the transit pose is only executed if a pick was found

Extended functionality for JavaScript

In order to implement complex functionalities even better, the JavaScript task has been expanded and the following improvements have been added:

- When renaming a Mikado variable, it is also renamed in *JavaScript* tasks
- XMLHttpRequests can be allowed optionally
- For picks from the picks search, both the TCP poses and the Tool0 poses are now accessible
- `lab.callService` allows the use of any ROS services
- Native robot programs can be called from JavaScript

Improvements for the pick search

The pick search has been expanded to include various new parameters and sorting criteria. In addition, feedback to the user has been improved:

- *Start handle search* now accepts a parameter for the (future) gripper
- **Position equality**: Separate parameters for tolerance in recognizing unsuccessfully grasped and re-recognized objects: `Part pose max. translation difference`, `Part pose max. rotation difference`, `Grasp max. translation difference`, and `Grasp max. rotation difference`
- **Sorting criteria:** New sorting criteria `Object angle to vertical` and `Number of pick attempts`
- **Pick inspector:** New button to center visualization to the middle of the box
- **Pick inspector:** Notify if transit pose is unreachable or other problems occured
- **Pick search:** Improved consideration of user binning

Improved traceability for fieldbus commands

To make it easier to track commands received via a fieldbus, these are logged on the new `Fieldbus` page. It can be found in Mikado under the `Run` menu item and provides information about the type of command received, start time, duration, and its status.

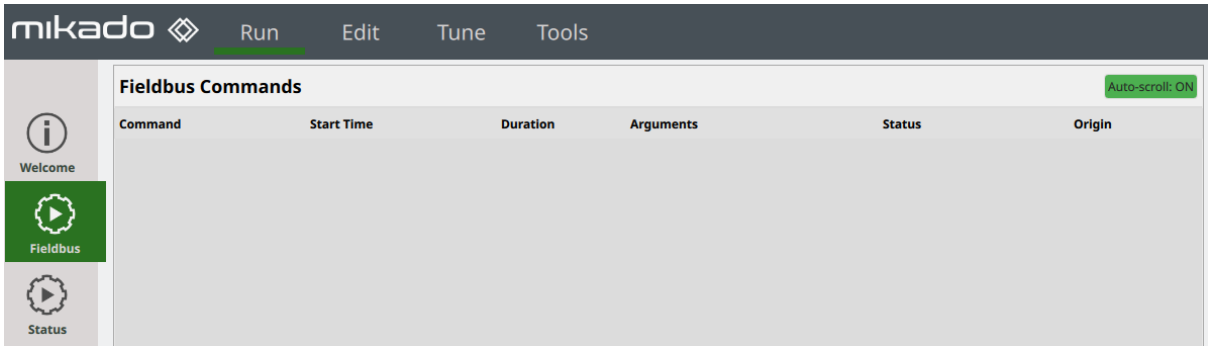

Fig. 18: All incoming commands are recorded and logged on the new fieldbus page.

Further fieldbus changes:

- **Fieldbus commands:** New commands for starting and evaluating a pick search directly via the fieldbus
- **Mikado Status:** Improved status messages regarding the loading of programs, products, and scenes

Object recognition

In version 3.3.0, the naming of symmetries has been revised and additional symmetries have been added. All symmetries are taken into account for object recognition and the `Move part to` task. On the recognition page, the new visualization of the views taken into account by PartFinder in the model generation tab, allow to better understand the effects of the symmetry selection.

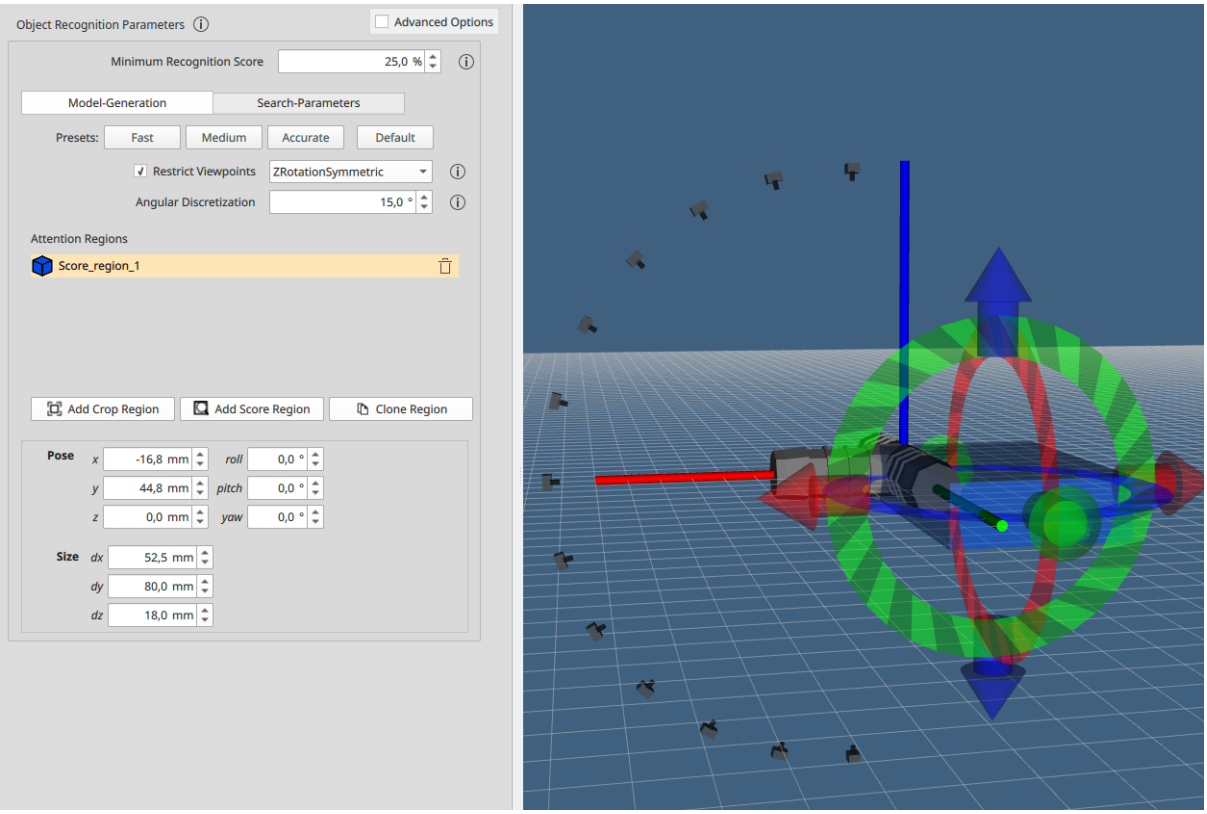

Fig. 19: New visualization of the views considered for object recognition and example of a `score region`

In addition, adjustments were made to Mikado to support Ensenso SDK version 4.2, resulting in a significant improvement in object recognition.

The following points have also been improved:

- **Save image:** The current product name can be used as a prefix when saving Filecams
- **Model generation:** Areas of the model can now be defined (`Score Region`) to be prioritized in the calculation of the recognition score
- **Product Mesh:** OBJ meshs can now be used directly as product meshes
- **Hypothesis clustering:** up to 4096 clusters allowed (as in Partfinder)

Debugging and Error Reporting

To simplify analysis in the event of an error, a button has been added to the Tools->Tools page to create an error report. After clicking on the button, you will be prompted to describe the error you have observed. After clicking on "OK," an archive file with relevant debug files is automatically created, which makes it much easier to investigate the error.

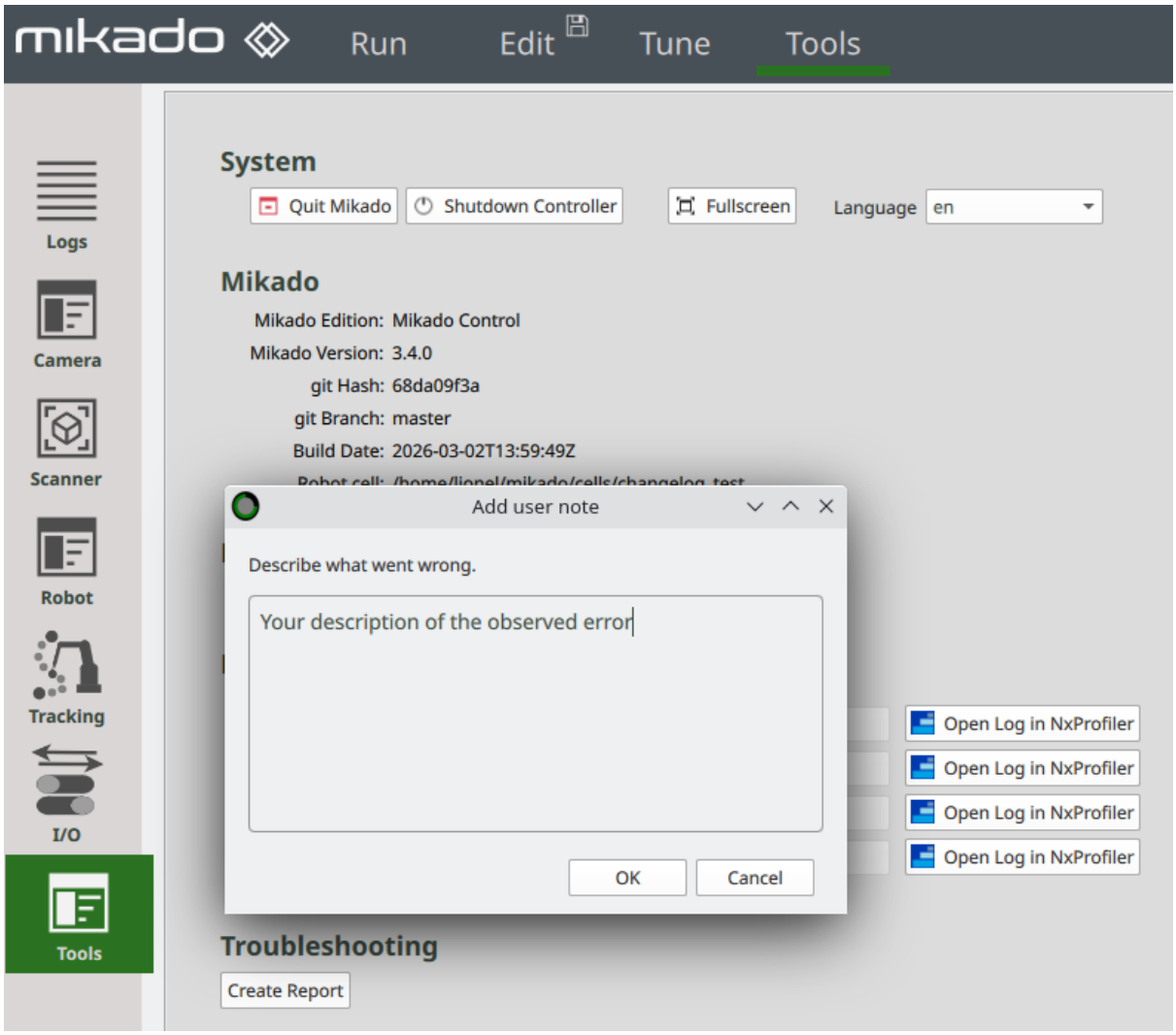

Fig. 20: Creating an error report on the Tools->Tools page

The log viewer under Tools->Logs has been modernized. The standalone log GUI can now also load saved logs for analysis.

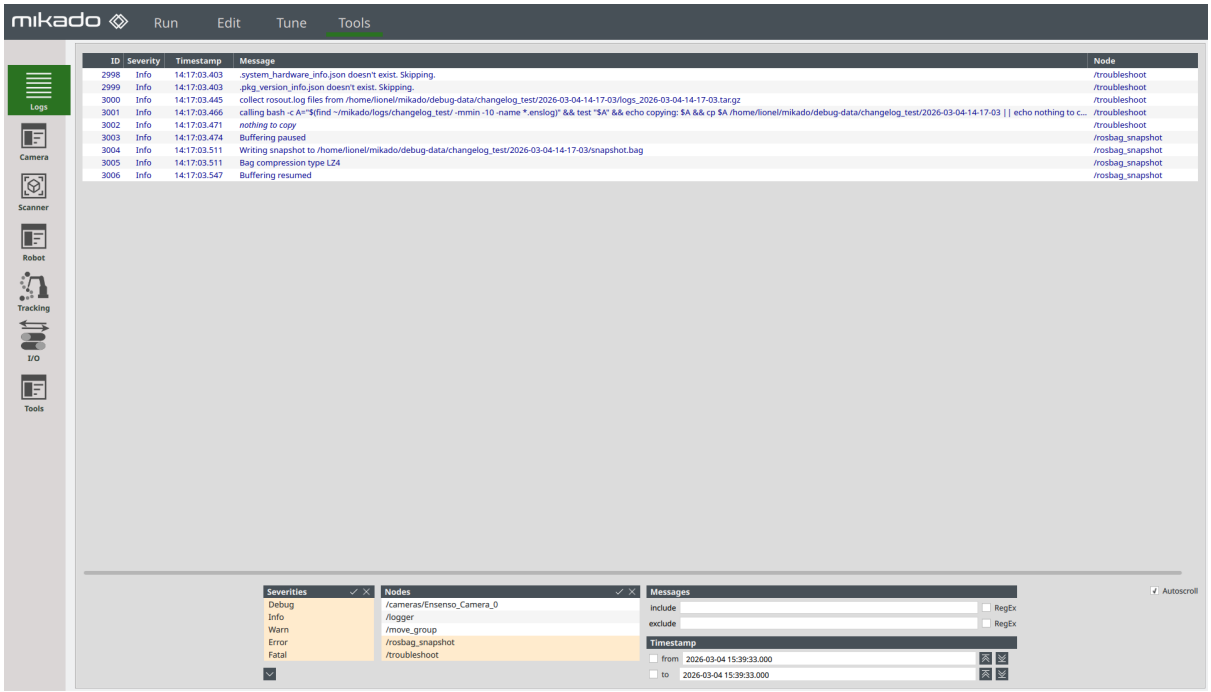

Fig. 21: View of the modernized Tools->Logs page

Better usability

- **Variables:** Cameras and grippers now have their own variable types
- **Grasp Editor:** Simplified setup with optional transparent products and markers adjusted to product size
- **Kiosk mode:** Exiting kiosk mode with Alt+Tab etc. is suppressed. The Windows/Meta key can also be suppressed using a KWin window rule. The status and logins in kiosk mode are logged.
- **Robot page:** The active gripper is now*displayed and can be changed
- **Setup mode:** New setup mode in which manual robot operation is not communicated as an error in the robot status
- **Scene objects:** Camera selection next to the capture button

Miscellaneous

- **Logging:** In addition to software versions, hardware and IP addresses are now also logged in the folder of a Mikado cell
- **Robot page:** Display of information about the robot model and manufacturer
- **Status bar:** Techman error codes are displayed in hexadecimal, as specified in the manual
- **Search strategies:** The step size of cartesian movements is adjustable
- **Kuka:** For Kuka robots, parameters for influencing speed have been added to the robot page
- **MeshScanner:** Do not always save meshes in OBJ format
- **Mesh names:** More special characters allowed in mesh file names
- **Program editor:** Warning if program cannot be written due to full hard disk
- **Scene objects:** Improved feedback when searching for box fails
- **Optimizations:** Optimizations in loading and validating programs as well as in calculating picks
- **Bug fixes:** Various bug fixes and stability improvements

1.4 Version 3.2.0

Graphical PickSearch tuning

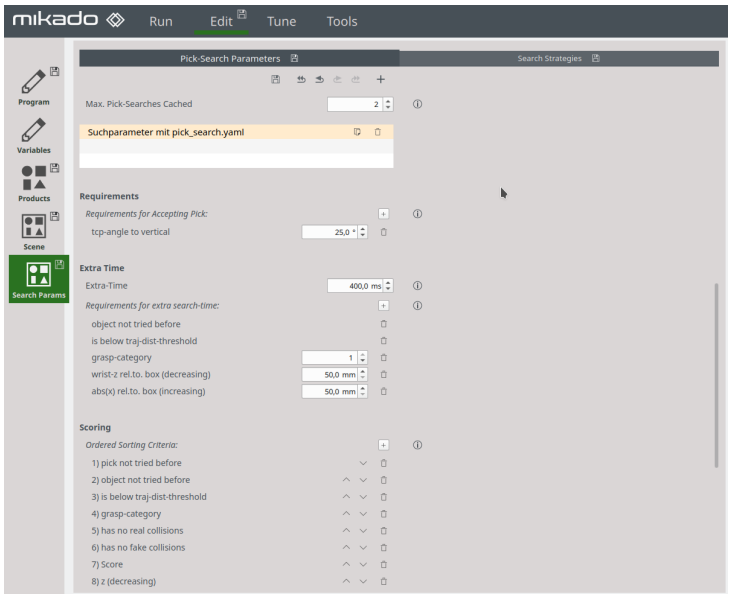

Fig. 22: Tuning the parameters of a pick-search

It is now possible to configure the parameters of a pick-search directly in Mikado. Depending on specific needs parameter sets can be set per product or switched on demand. Use cases may include enforcing a strict order of depalletizing or prioritizing certain extraction strategies during a pick. The rudimentary configuration options that previously existed in Mikado Assistant have been removed.

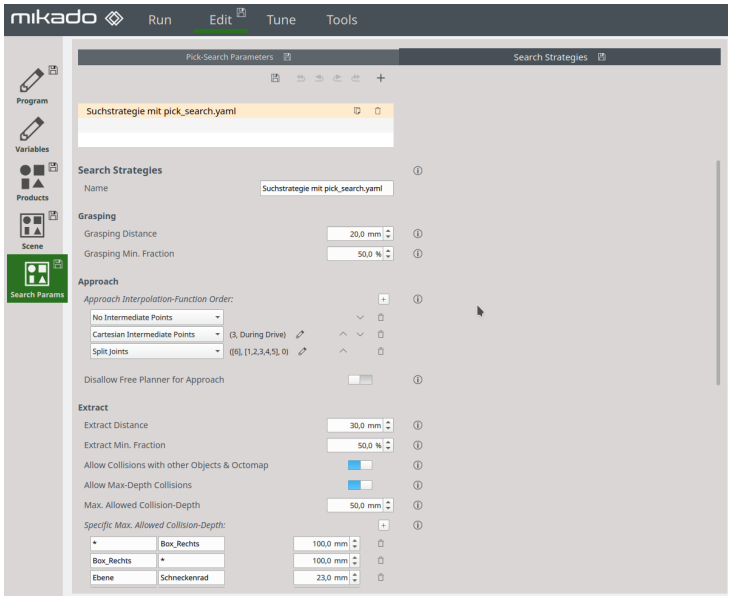

Fig. 23: Configuring the extraction strategies

Some new extraction strategies were added as well. Especially sliding out objects in their xy-planes has proven valuable for randomly stacked cut-metal parts.

Switching between several pick-searches

Multiple pick-searches can now be switched between to allow efficient alternating picks from multiple boxes. All pick-searches can be inspected and optimized using the pick inspector or best picks tool. Multiple grippers can be tested without requiring a new object search. This happens either automatically when rerunning Start Pick-Search or by using JavaScript (see *rerunPickSearch*).

Numeric IDs for grippers, cameras, products and scene objects

To simplify external control of the process flow using a PLC it is now possible to assign numeric IDs in addition to the existing names. This was added for grippers, cameras, products and scene objects.

Improved communication of joint limits

Issues due to violated joint limits are now more readily visible. Error messages were added at program start, when jogging or in the editor for joint position variables.

New PartFinder Features

Some new parameters of the PartFinder object recognition algorithm added with Ensenso SDK 4.1 were made accessible. This includes TexturePenalty or ComplementViewpoints. Additionally it is now possible to restrict the object recognition to certain regions of the model without artistically editing the CAD mesh model. These so called attention regions can be configured intuitively now.

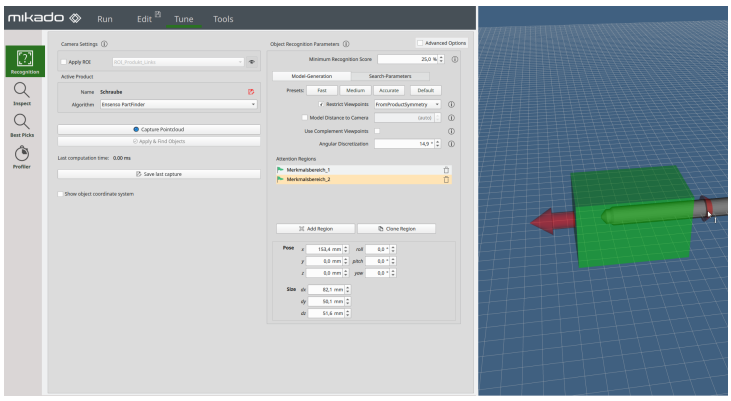

Fig. 24: Example attention region: only the tip of a screw is of interest

For rapid interpretation of object recognition results it is now possible to display their local coordinate axes.

Accessing Picks in Mikado Select

The collision-checked pick search in Mikado Select is now performed asynchronously as in Mikado Control via Start Pick-search and the new Get Picks task. The previous Find Objects task is automatically replaced by the two tasks during the upgrade if the pick-search was previously used.

Further innovations

- Program: Tasks can now be disabled or commented out and are skipped during execution
- Jogging: Filter variable names
- Filter variables in variable picker dialogue
- JS: reading and writing IOs (see *robot*)
- JS: set the state of a task to Error or ErrorMightRetry (see *task*)
- JS: lab.callExtensionService for arbitrary extension scripts (see *callExtensionService*)

- WaitForFieldbusInputSignalTask: task to wait for value changes in a (fieldbus) variable (see *Wait for fieldbus input signal*)
- SetInitialValueTask: optionally save the program (see *Set initial value*)
- Display variables from "addons" as well
- Status: jump to currently active task
- Yaskawa specific functions and error codes
- FindObjectTask no longer needlessly cancels pick-searches running in parallel
- Number of contact points in collision checking can be configured
- Default number of contact points during collision checking was increased to get more realistic overlap results
- PickInspector: more details why picks where filtered
- PickInspector: communicate clearly when grasps collide with the object to be grasped
- Variable page now also displays when variables are used in JavaScript tasks
- Show the number of triangles in a scene objects mesh
- Immediately cancel asynchronous capturing if the robot fails to move
- Use new profiler file format
- separate package mikado-robots-common for helper programs
- Usability improvements in the grasp editor
- Filecams are now stored in mikado/filecams and are limited in size
- New task: Reset scene object position

1.5 Version 3.1.2

References to scene objects and products

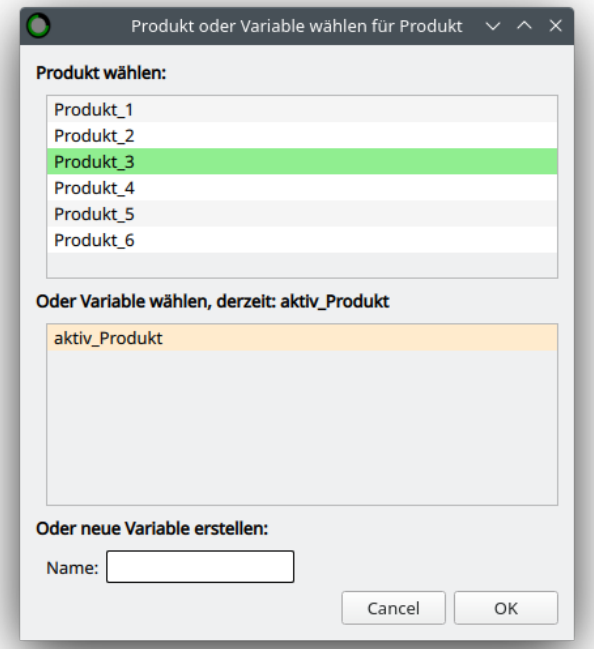

Fig. 25: Selection of a program variable or a product reference

Products and scene objects can be referenced directly in tasks. This feature is particularly useful if the same ROI is always to be used within a task or the same product is to be searched for. It is still possible to set the value via a program variable.

Revision Robot tracking

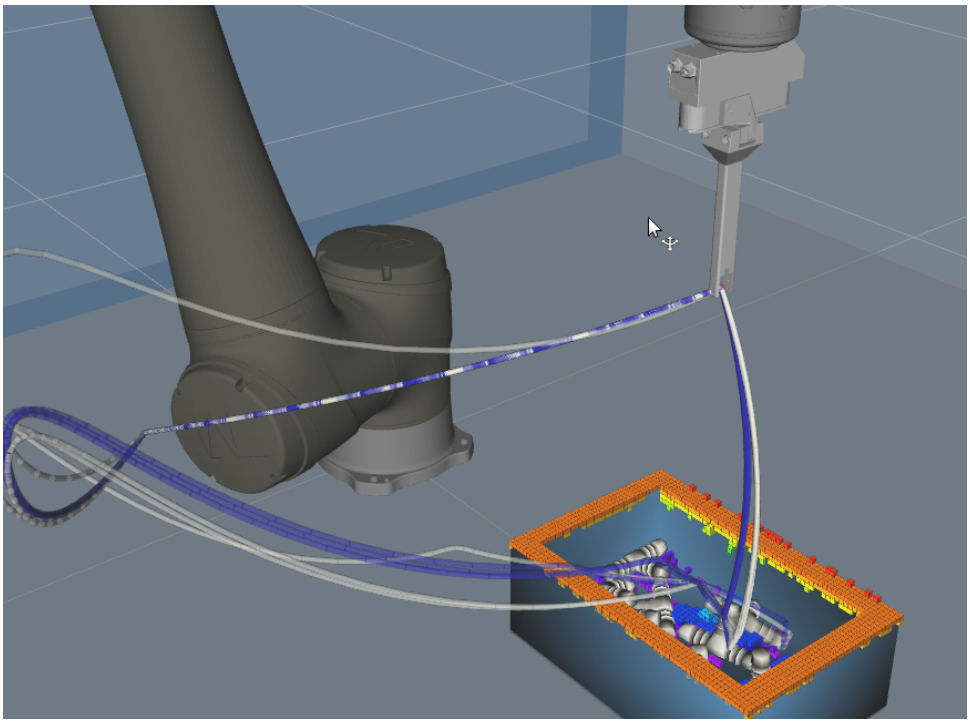

Fig. 26: Visualized path of the active TCP

On the *Robot tracking* page, the path of the active ToolCenterPoint (TCP) is visualized as a line. The page also supports the use of multiple grippers.

Position controlled grippers

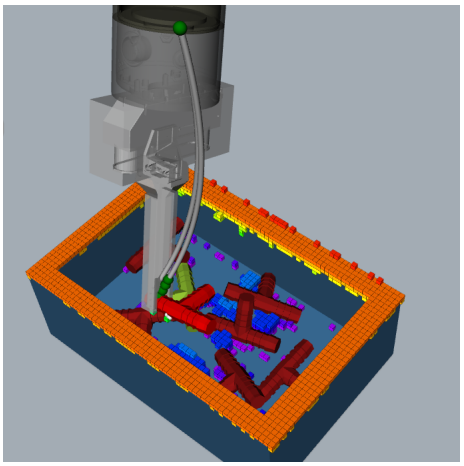

Fig. 27: Grasp with small opening width

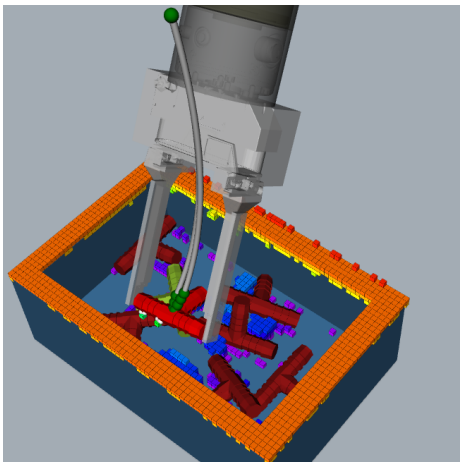

Fig. 28: Grasp with large opening width

For grippers with position control, the opening width of the gripper fingers can be set for each Grasp-Setting. To do this, the gripper type must be set to `type: ParallelPositionControl` in the corresponding `gripper.yaml`.

Mesh-Scanner

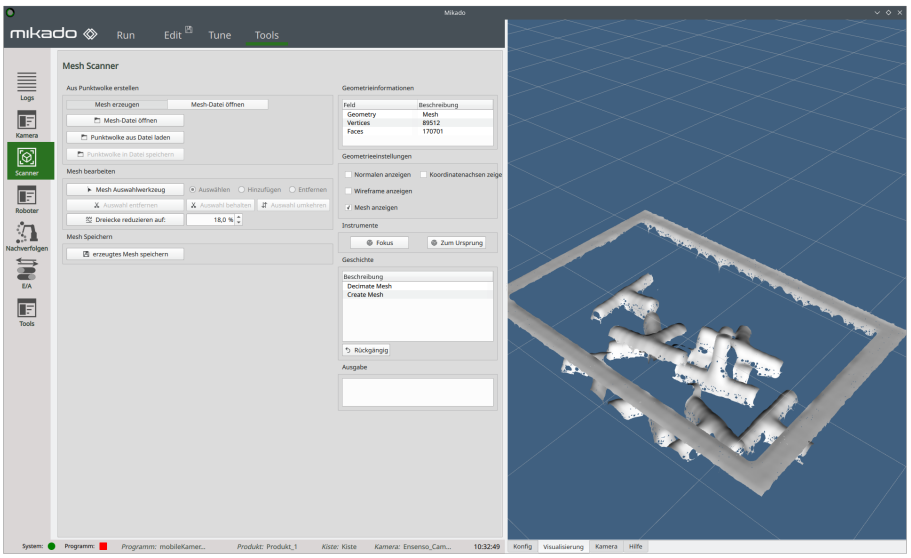

Fig. 29: Screenshot Scanner Page

The scanner page offers the option of creating a mesh from a point cloud and post-processing the created mesh. The mesh can then be used in the product search. The scanner is able to:

- Capture point cloud or load point cloud/mesh
- Create mesh from point clouds
- Simplify mesh
- Cut out parts of the mesh
- Save mesh

Collision check of grasps in the grasp editor

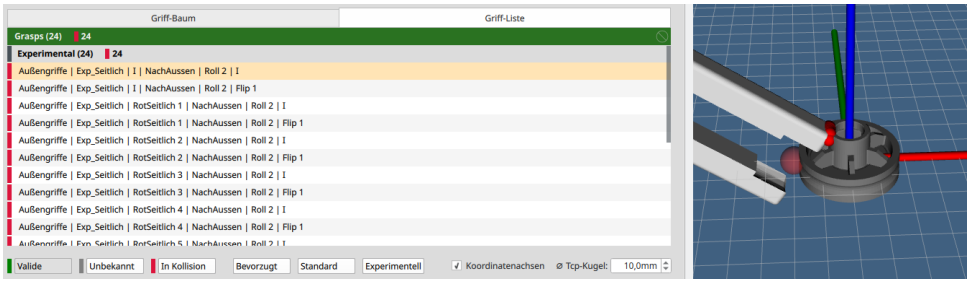

Fig. 30: Grasp list and visualization of collision points

Newly created grasps are checked for collisions with the part when they are created (see *Collision Check for Grasp-Poses*). Next to the grasp tree there is the tab grasp list. This list contains all grasps generated by the grasp tree. If you select a grasp in the grasp list, it is displayed in the visualization. If a grasp collides, the detected collision points between the gripper and the product are marked with small red spheres.

Minor optimizations

- Significantly faster Profinet interface thanks to event-based backend. For Details refer to: *Version 3.1.0*
- GUI language can be changed without restarting Mikado
- Robot status display tidied up, more error details
- Status and program editor: more informative placeholder texts
- Update Octomap now also possible from JS, without pick search / FindObjects (see *updateOctomap*)
- Pick inspector: pick search can be started manually
- Picks can be sorted by *Collision-Depth*
- JavaScript: Read CSV file into dictionary, e.g. for product parameters (refer to *file*)
- Warning if a product uses very large meshes
- Erroneous variables are displayed on the variable page, e.g. writing to read-only variables or variables that have an incorrect type when used
- JavaScript: Tcp coordinates for grasps available, i.e. Start, Approach, Retract and Final (siehe *picksearch*)
- QML: it is now possible to start any process, e.g. Okular for a project specific manual
- Interface to implement object recognition with a project-specific algorithm
- PickSearch/JS: Additional freely usable evaluation criterion for objects (user binning)
- Adjusting the position of a gripped part
- Move robot to: possible collisions directly in the status of the task
- Grasp editor: Copying and moving grasps
- X > Y? and similar tasks with blue symbols for logic/jumps
- Task to read out the length of a list
- FindObjects: optional sequence change if box was searched for and not found
- JS: More information on last used grasp
- Picks can be sorted by collision depth

1.6 Version 3.0.5

- Error corrections in the handling of newly created variables

1.7 Version 3.0.4

- Cuda can be activated for the PartFinder product search
- Very short Cartesian moves are no longer extended to 10 points

1.8 Version 3.0.3

- For developers, “Find Objects” is also visible in Mikado Control
- Find Objects now also has an optional “Box” parameter
- PartFinder: maximum number of clusters increased to 200
- new Task: Move Tool Center Point (TCP) Cartesian, optionally with tolerance to the target.
- Display information about the currently used FileCam on the object detection page
- Grasp translations can also be set to symmetric now

1.9 Version 3.0.2

- Picks: entire trajectory can be visualized in preview
- read-only variables
- Variable page: favorite variables

1.10 Version 3.0.1

- Detailed display of object search / pick search pipeline
- ADS interface support
- NxLogs GUI for profiling

1.11 Version 3.0.0

Collision check of the gripper for Mikado Select

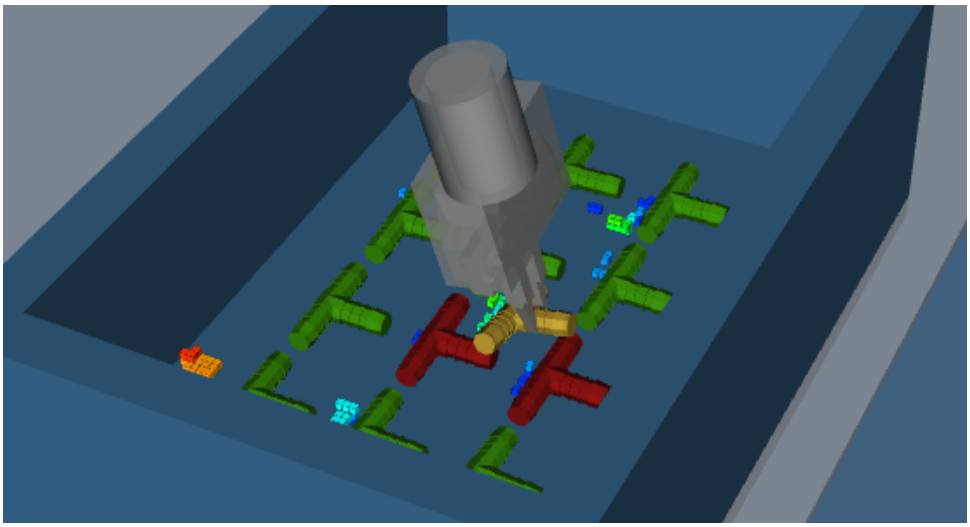

Fig. 31: Collision check of the gripper for Mikado Select

Mikado Select supports the finding of collision-checked handles. The task *Find object* has a new return value `Valide Griffe` for this purpose. This pose list contains the best grasp per found and graspable object. The list is sorted according to user-specific criteria (see *Best handles*).

Visualization of the handle prioritization

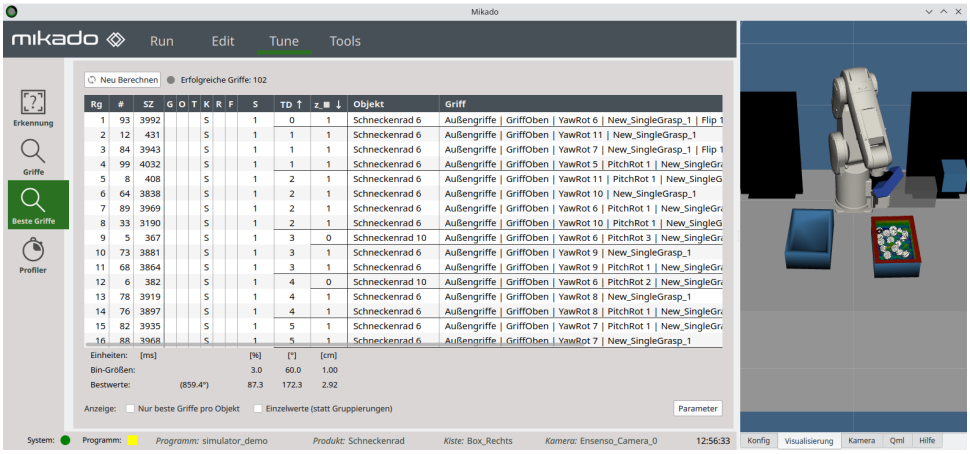

Fig. 32: Screenshot Best-Griffe

Sorting and exclusion criteria for handles can be defined globally (for all products) in `pick_search.yaml`. Binning is used to compare all handles according to the various criteria. The new *Best handles* page provides an analysis tool for the sorting criteria.

Support for tool changers

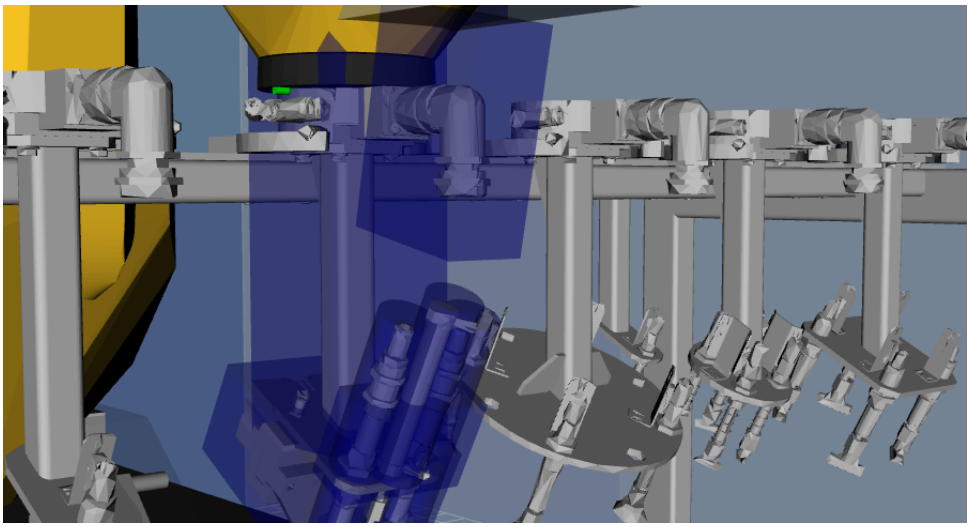

Fig. 33: Support for tool changers

Mikado Control supports changing grippers during the runtime of a robot cell. The new task *Set active gripper* can be used to switch between the defined gripper geometries within the robot program. Products now have gripper-dependent grip definitions, see *Setting up handles for interchangeable grippers*. The grip search only checks grips that have been defined for the actively set gripper.

Setting up a cell with multiple grippers will be handled by Mikado support for the time being. The addition of new grippers in Mikado Assistant is in progress and is planned for the next release. Editing cells already set up in Mikado Assistant is fully supported.

Additional joints also for parallel grippers

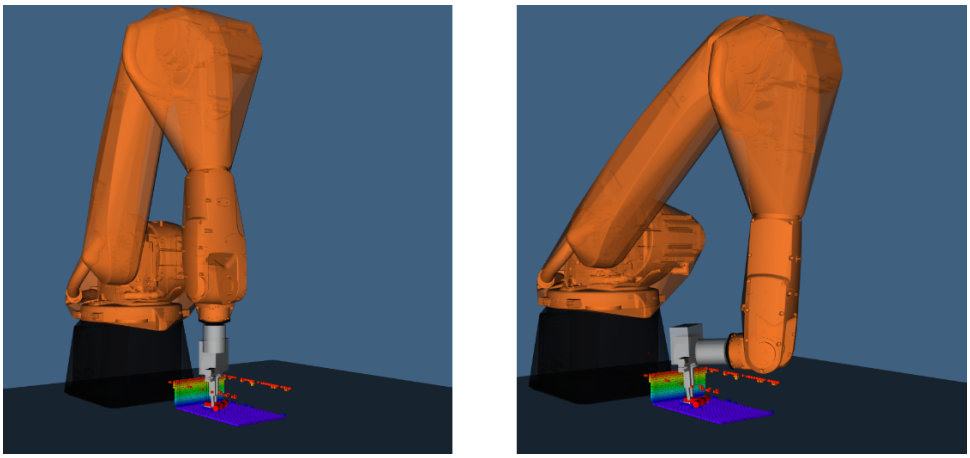

Fig. 34: Tilting joint for parallel gripper

Additional joints on the gripper are also supported for parallel grippers. Previously, only magnetic and vacuum grippers were possible (geometry does not change when opening/closing). Further details can be found in the documentation of the *Setting the gripper status* task.

Library with sample programs

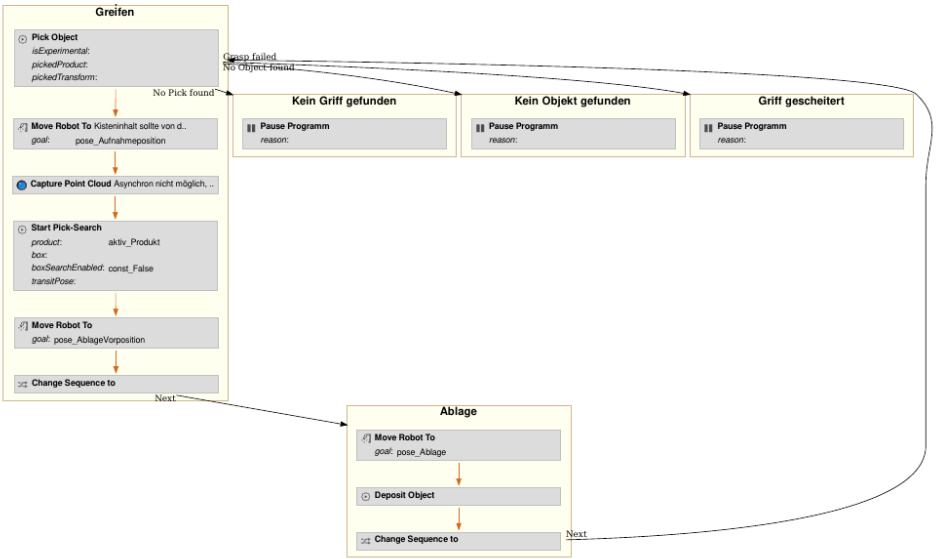

Fig. 35: Sample program for mobile camera

The program library is located on the Welcome page and can be opened by clicking on the folder icon to the left of the 'Programs' heading. It offers sample programs that are suitable as a starting point for project-specific programs. These include:

- **Select**
  - TCP calibration program
  - Finding objects and transferring the results via Profinet interface
- **Control**
  - TCP calibration program

– Asynchronous image recording with stationary camera
– Synchronous image recording with mobile camera

If-Else tasks for conditional execution of partial sequences

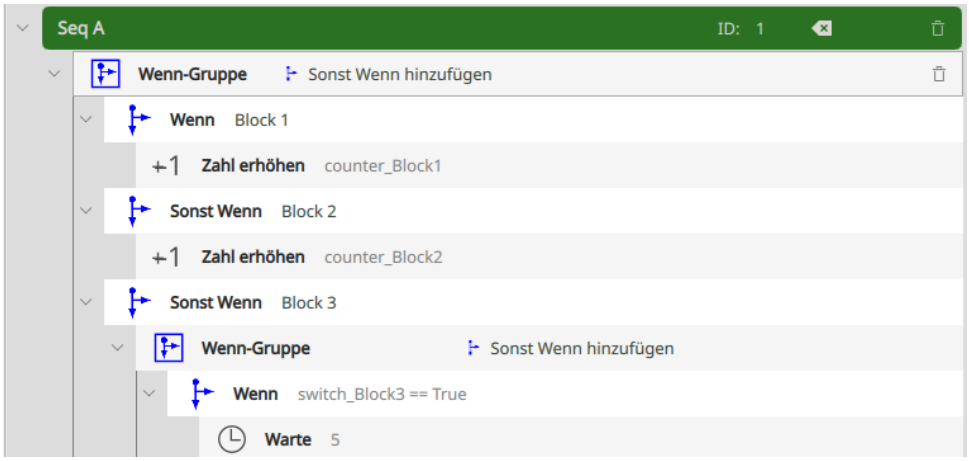

Fig. 36: Nested if-groups in the program editor

The new *If-Group* task enables the conditional execution of partial sequences. This feature also enables, among other things:

- Copying sequences
- Subdividing sequences into partial sequences
- Parallel deletion of multiple tasks
- Redesign of the program editor

Rotation on jogging side

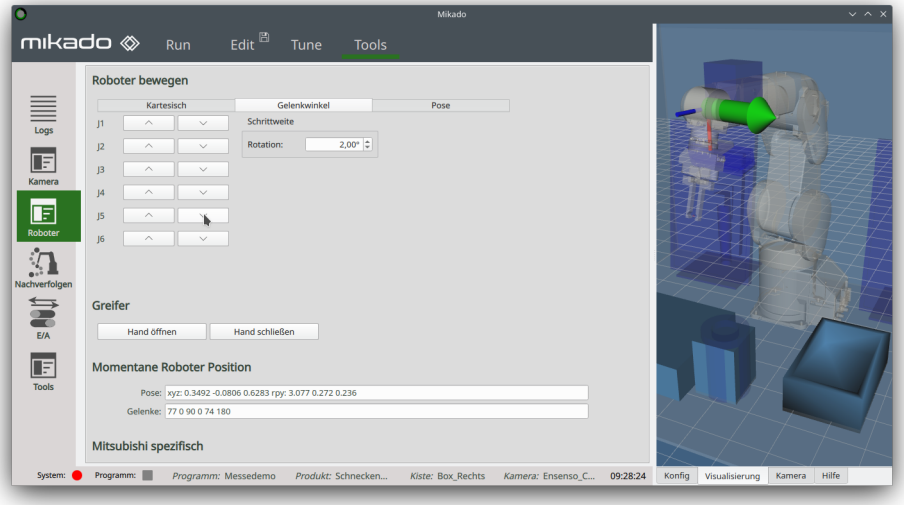

Fig. 37: Jogging page: Preview of the rotation axis with green arrow

In addition to Cartesian movements in the x, y and z directions, the robot can be rotated piece by piece on the robot page (see *Robot control*). The coordinate system in which the robot is rotated can be selected manually. The

following coordinate systems are available:

- World - Coordinate system
- Robot - Coordinate system
- Tool0 (or robot wrist) - Coordinate system

Further innovations:

- The movement is displayed as a preview on the right-hand side
- Redesign of the robot page

Depalletization rules

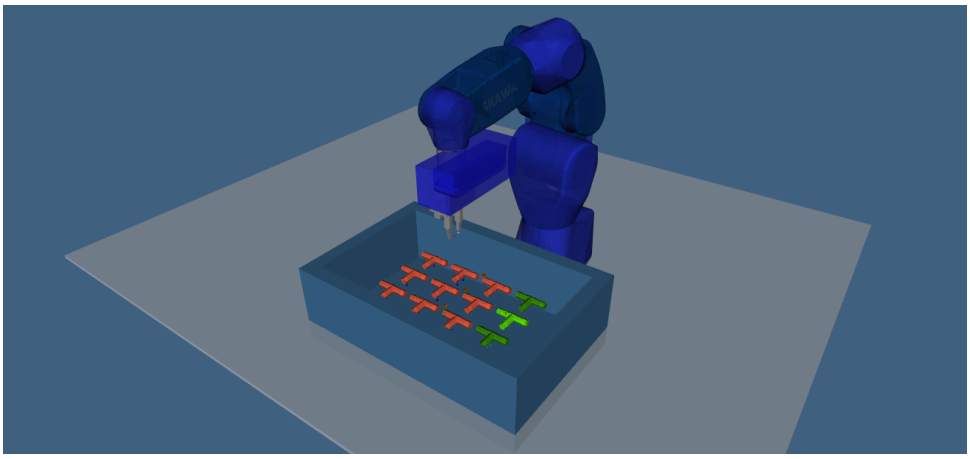

Fig. 38: Only picks with the smallest x-value are permitted

The new parameter `max_bins_required` enables the exclusion of certain findings (see *Conditions*). For example, products with a high x-value relative to other products can be excluded. This enables depalletizing from left to right.

More information on failed picks

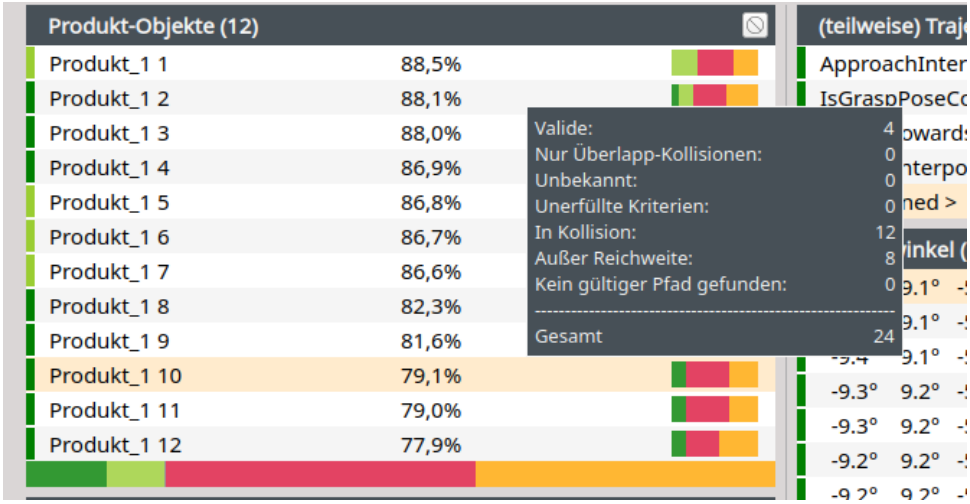

Fig. 39: Re-design of the Pick Inspector page

The pick inspection page has been revised and now offers the following additional features:

- Some collision pairs have additional explanatory text
- Bar charts show the current status of the handle search as a whole and the product-specific status

JavaScript: API for scene objects

```
var size = scene.getSceneObjectDimensions("ROI1")
scene.setSceneObjectDimensions("ROI1", size.x, size.y, 0.23)
```

Fig. 40: Setting the z-value of an ROI to a fixed value

The size and position of scene objects can now be read and written via the JavaScript API (see *scene*). The following new functions are available:

- `getSceneObjectDimensions`
- `setSceneObjectDimensions`
- `getSceneObjectPose`
- `setSceneObjectPose`

1.12 Version 2.1.0

- Integration of Ensenso PartFinder object recognition
- Object detection parameters are saved and set on a product-specific basis
- Total number of handles is displayed with the product
- Both individual grip patterns and all resulting grips can be displayed

Pick inspection:

- Objects are colored according to their tangibility and can be clicked on directly
- Picks can be filtered by success

Extensions:

- The results of the handle search can now be called up via JavaScript
- JavaScript kinematics API to check the accessibility of multiple poses or paths, for example
- The dimensions of the point cloud within an ROI can be queried via JavaScript
- Calibrations via a ROS interface (and thus e.g. via OpcUA or ProfiNet)

2 Scene objects

2.1 Version 5

Warning

If your program uses a *Read scene object position* or *Set scene object position*, the behavior of the program may change. Please check your program accordingly.

The meaning of the `default pose` of a scene object has been changed. In version smaller than 5, the default pose describes the center of the floor of a scene object. The `default size` parameter was used to determine the floor. This meant that the actual pose of the scene object depended on the default pose and the default size. The dependency on the default size has been removed. As of version 5, the default pose for scene objects with cuboid

geometry (such as ROIs) describes the position of the center of gravity of the volume instead of the center of the floor as before. For scene objects with mesh geometry (such as boxes), the default pose now directly describes the position of the origin of a mesh in the world coordinate system. In addition, the new parameter `Mesh-Offset` has been introduced to be able to move the origin of a mesh from Mikado.

With the version update, the various pose parameters for scene objects are automatically changed to the new meaning. The following operation is performed:

1. Shift the `default pose` by half the z-component of the default size d_z in the positive z-direction of the local coordinate system,
2. Shifting the `transition pose` by the same amount and in the opposite direction and
3. Set the z-component of the `mesh offset` to $-\frac{1}{2}d_z$ (same amount as for 1 and 2).

The new meaning of the default pose is shown in the following illustration.

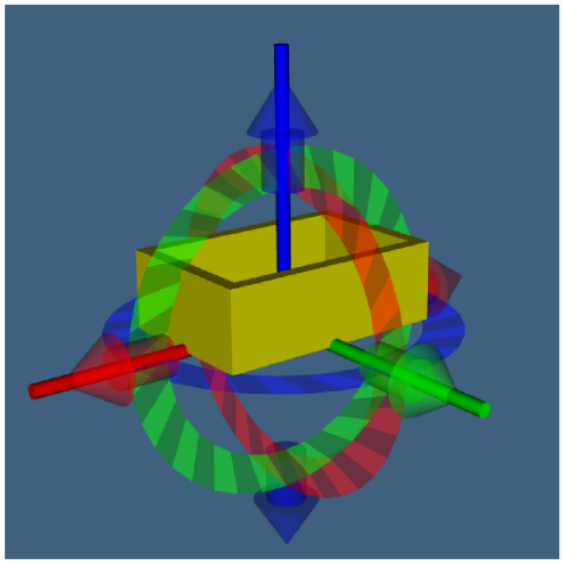

Fig. 41: Old meaning of the `default pose`

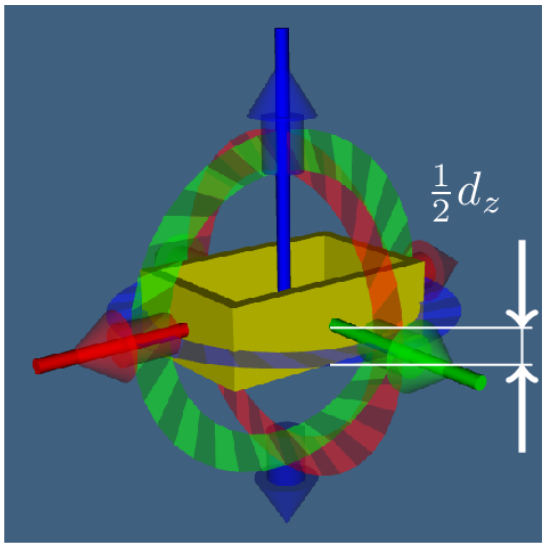

Fig. 42: New meaning of the `default pose`

The default size is set in Mikado as shown in the following screenshots.

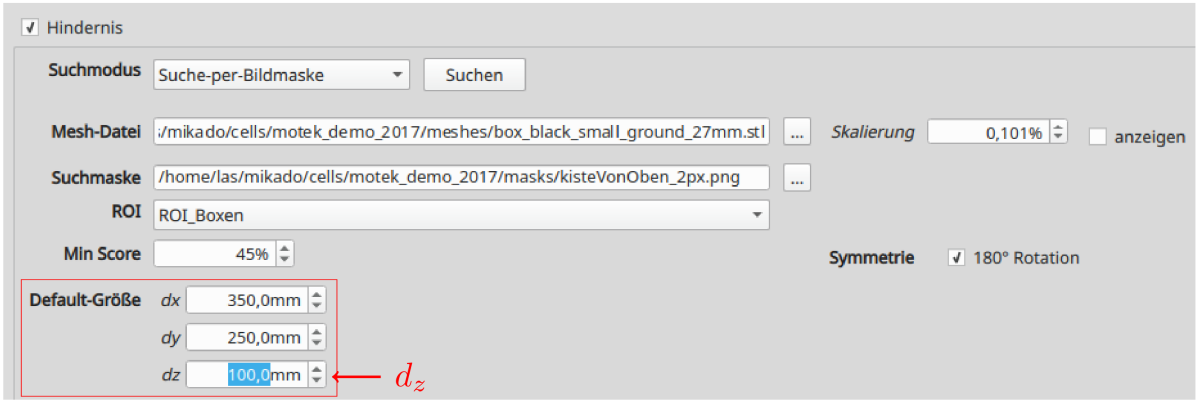

Fig. 43: Setting the default size of a crate

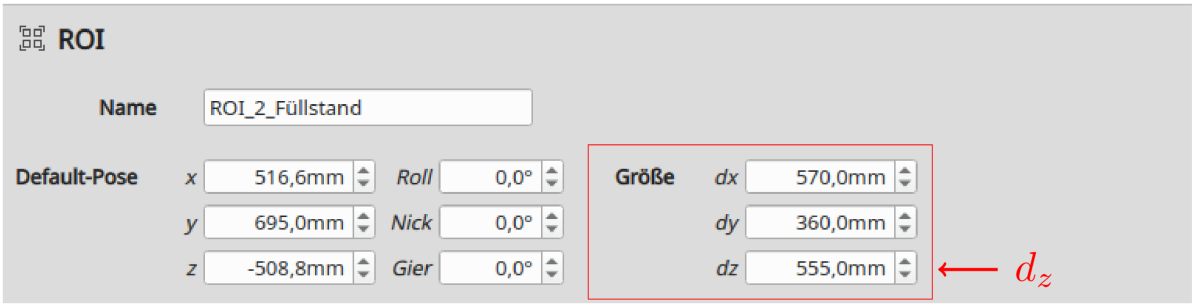

Fig. 44: Setting the default size of an ROI

2.2 Version 4

New function `ROI` added, which is required when searching for scene objects. Please select an ROI object for each scene object that uses `Search-By-Image`.

2.3 Version 3

New function `Search symmetries` added, which is used when searching for scene objects. Disabled by default.

2.4 Version 2

The standard file extension for scene objects has changed. Please save your Mikado scene object files with the extension `*.scene` from now on. The paths for the start program in config/mikado.yaml must be edited manually.