optonic
vision · automation

Aktualisierungen

mikado

Version 4.0

Ein Produkt von 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

Inhaltsverzeichnis

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 . . . 17
- 1.5 Version 3.1.2 . . . 19
- 1.6 Version 3.0.5 . . . 23
- 1.7 Version 3.0.4 . . . 23
- 1.8 Version 3.0.3 . . . 23
- 1.9 Version 3.0.2 . . . 24
- 1.10 Version 3.0.1 . . . 24
- 1.11 Version 3.0.0 . . . 24
- 1.12 Version 2.1.0 . . . 29

2 Szenen-Objekte **29**
- 2.1 Version 5 . . . 29
- 2.2 Version 4 . . . 31
- 2.3 Version 3 . . . 31
- 2.4 Version 2 . . . 31

In diesem Kapitel werden Änderungen in Mikado-Versionen detailliert beschrieben. Die Liste der Änderungen an allen Mikado Komponenten finden Sie auch immer aktuell auf unserer Webseite unter https://optonic.com/support.

1 Mikado

Untenstehend finden Sie die Änderungen für alle Mikado Version.

1.1 Version 3.5.0

Mikado 3.5 erweitert die Planung beim Herausnehmen von Teilen aus der Kiste. Neue Zeitmessungen und Fehlerberichte unterstützen die Optimierung von Programmabläufen; weitere Verbesserungen erleichtern das Einrichten und Bearbeiten von Zellen.

Rückfahrt in Abschnitten planen

In Mikado Control lässt sich die Rückfahrt nach der Entnahme mit der Interpolationsart `In Abschnitten` gezielt aufteilen. So kann das gegriffene Teil beispielsweise zuerst parallel zu den Kistenwänden ausgerichtet und anschließend gerade nach oben gefahren werden. Die Abschnitte werden in der eingestellten Reihenfolge geplant.

`Zur Kiste ausrichten` bietet Zielwerte und Toleranzen für Position und Orientierung des Teils relativ zur Kiste. `Gerade nach oben fahren` führt das Teil entlang der positiven Z-Achse des Kistenkoordinatensystems bis zum gewählten Z-Zielwert. Unter der Abschnittsliste legen Sie fest, ob die Planung anschließend bis zur Transitpose weitergeführt wird.

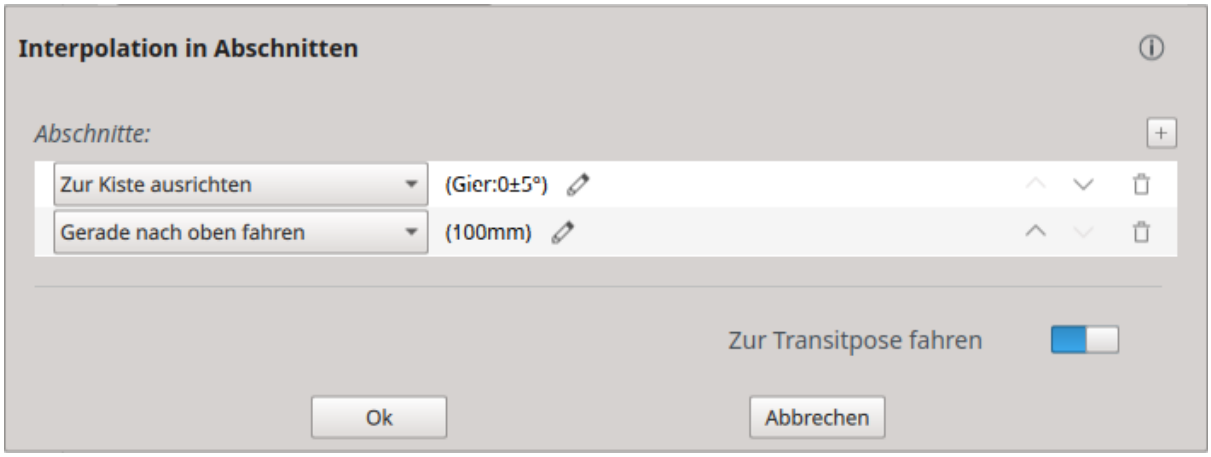

Abb. 1: Die Abschnittsliste legt die Reihenfolge fest; die Fahrt zur Transitpose ist separat wählbar.

Die einzelnen Einstellungen sind unter Interpolation in Abschnitten beschrieben.

Kollisionen bei der Entnahme gezielter erlauben

Die Suchstrategie unterscheidet jetzt Kollisionen des gegriffenen Teils mit anderen Funden, mit der Punktwolke (Octomap) und mit der Kiste. Für jede dieser Gruppen können Sie unabhängig wählen, welche Kontakte erlaubt sind und bis zu welcher Kollisionstiefe.

`Keine` verbietet Kollisionen dieser Gruppe. `Greifpunkt` erlaubt nur Kontakte, die bereits am Greifpunkt bestehen, und nur bis sie erstmals verschwinden. Ein späterer erneuter Kontakt ist damit nicht erlaubt. `Alle` erlaubt auch neue Kontakte innerhalb der eingestellten Tiefe. So lassen sich beispielsweise kleine Überschneidungen aus der Objekterkennung gezielt tolerieren.

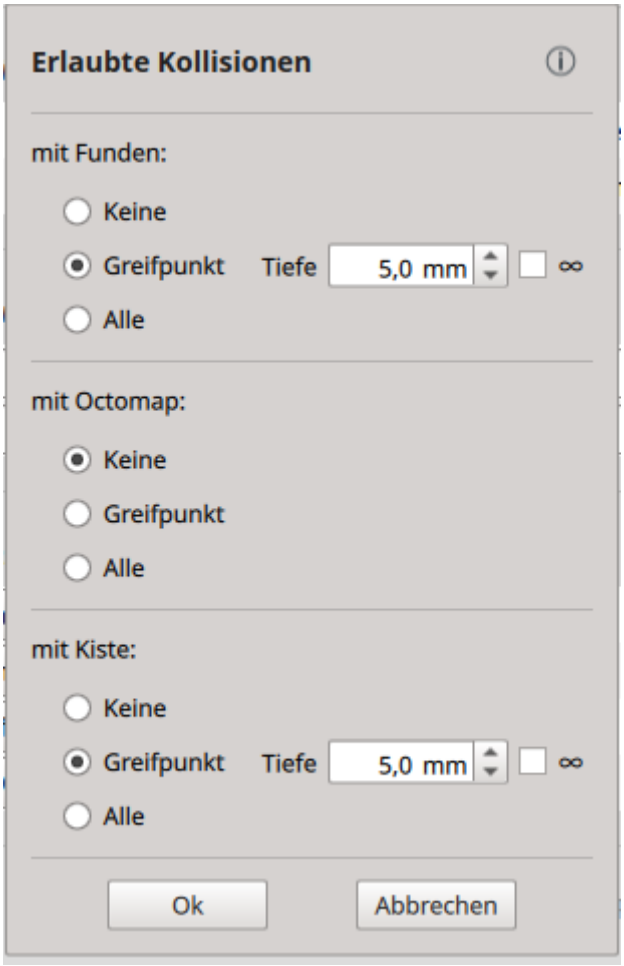

Abb. 2: Jede Kombination legt die erlaubten Kollisionen für alle drei Gruppen fest.

Mehrere Kombinationen können als Alternativen angelegt werden. Die Griffsuche prüft jede Entnahmefunktion mit jeder Kombination. Die Griffinspektion zeigt dazu detailliertere Kollisionsangaben und Gründe für nicht erfüllte Entnahmebedingungen. Weitere Informationen finden Sie unter *Entnahme*.

Zykluszeiten mit der Stoppuhr messen

Der neue Task *Stoppuhr* misst die Dauer eines Ablaufs direkt im Mikado-Programm. Verwenden Sie vor dem zu messenden Ablauf den Modus `Start / Runde` und danach `Stop`. Beide Tasks müssen denselben Wert für `Label` verwenden, beispielsweise `PickCycle`.

Die optionale Ausgabe `Intervall [s]` enthält die gemessene Zeit in Sekunden. Für wiederkehrende Abläufe können Sie stattdessen an derselben Stelle jedes Zyklus erneut `Start / Runde` aufrufen: Der erste Start gibt `0` aus und alle folgenden Aufrufe messen die Zeit seit dem vorherigen Aufruf. `Median [s]` liefert den Median der letzten bis zu 1000 erfassten Intervalle dieses Labels.



Abb. 3: Über dasselbe Label gehören Start, Rundenmessungen und Stopp zu einer gemeinsamen Stoppuhr.

Fehlerberichte passend zum Anlass erstellen

Auf der Systemseite können Sie beim Erstellen eines Fehlerberichts entscheiden, ob die letzte Kameraaufnahme als Filecam und ein vollständiges Zellbackup enthalten sein sollen. Ergänzen Sie eine Notiz, die den beobachteten Fehler und den Zeitpunkt beschreibt. Die Filecam hilft, die zuletzt aufgenommene Szene später erneut zu untersuchen.

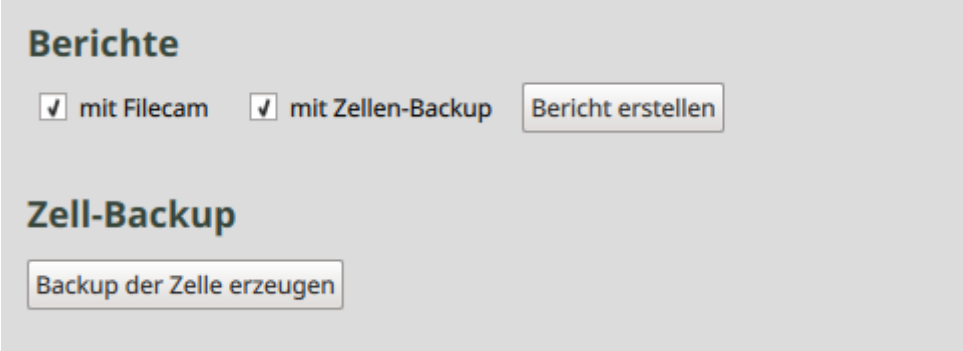

Abb. 4: Die zusätzlichen Berichtsinhalte lassen sich einzeln auswählen.

Mit dem neuen Task *Fehlerbericht speichern* kann ein Programm einen Bericht beispielsweise in einer Fehlersequenz selbst erstellen. Optional werden die letzte Punktwolke und eine Notiz zum Anlass hinzugefügt; der Task-Kommentar wird ebenfalls aufgenommen. Das Erstellen benötigt Zeit und eignet sich deshalb nur für gezielte Diagnosefälle.

Zellbackups werden jetzt als `tar.gz`-Archive mit Zellname und Zeitstempel gespeichert. Ist `pigz` verfügbar, nutzt die Komprimierung mehrere Prozessorkerne. Der eigenständige Logger kann außerdem übergebene Logdateien und ganze Logordner direkt öffnen.

Toleranz-Parameter für 2D-Kistensuche

Die 2D-Kistensuche kann ungültige Funde unter Angabe einer maximal erlaubten Positions- und Winkelabweichung auswerten, siehe *Kistensuche*. Wird einer der beiden Grenzwerte überschritten, gilt die Suche als fehlgeschlagen.

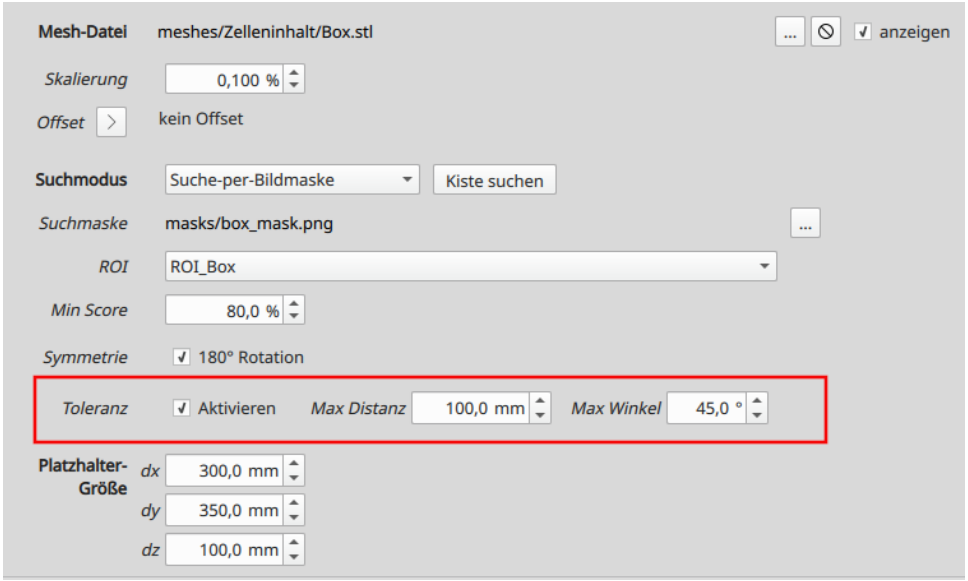

Abb. 5: Toleranz-Parameter für 2D-Kistensuche

Produktgeometrie aus parametrischen Vorlagen erzeugen

Produkt-Geometrien können optional aus vorgegebenen parametrischen Vorlagen erzeugt werden, anstatt eine Mesh-Datei vorzugeben. Dafür stehen die geometrischen Formen Quader, Kugel, Zylinder und Hohlzylinder zur Verfügung. Die Parameter der Vorlagen lassen sich sowohl über die Benutzeroberfläche als auch via JavaScript im Programmablauf ändern.

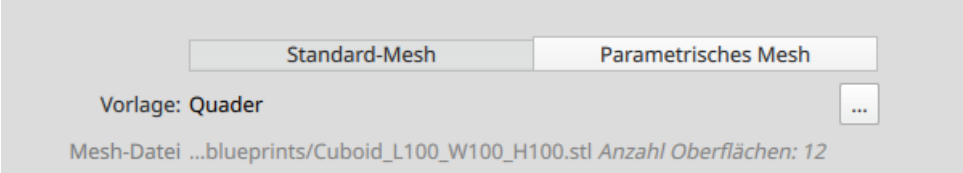

Abb. 6: Auswahl der parametrischen Vorlage in den Produkt-Einstellungen

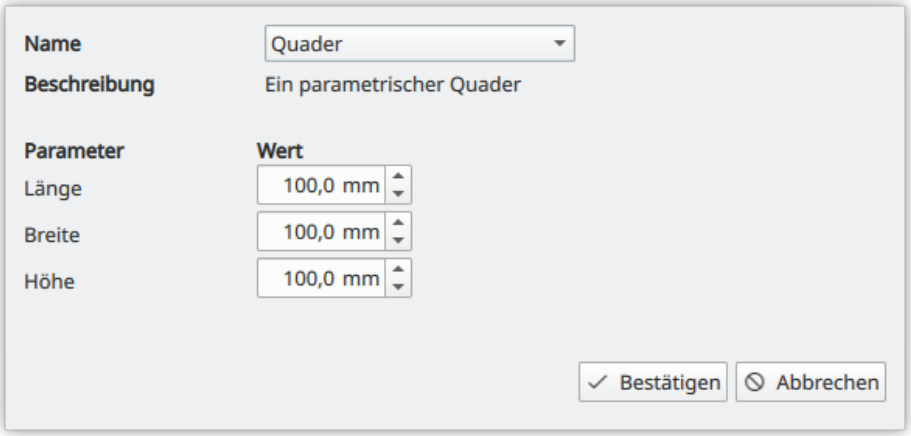

Abb. 7: Dialog zur Bearbeitung einer parametrischen Vorlage

Weitere Verbesserungen

- **Programme:** Neue Standardabläufe für Mikado Control und Select sowie ein Beispiel für die Hand-Auge-Kalibrierung mit festen Posen wurden ergänzt. Diese sind in der Programm-Bibliothek auf der Willkommen-Seite zu finden. Geladene Beispielprogramme öffnen sich außerdem nun direkt im *Programm-Editor*.
- **Laden von Dateien:** Beim Öffnen externer Programm-, Produkt-, Szenen- oder Meshdateien können diese automatisch in die Zelle kopiert werden.
- **Kalibrierung:** Vor der automatischen Hand-Auge-Kalibrierung müssen der aktive Greifer und die passende Szene gewählt werden, um Kollisionen durch unachtsam gewählte Einstellungen zu vermeiden. Die Bewegungen begrenzen die Geschwindigkeits- und Beschleunigungsskalierung auf höchstens 10%. Es wird keine freie Pfadplanung verwendet.
- **Griffe und Griffinspektion:** Kamera, ROI, Parametersätze und Suchstrategien sind bei der ausgewählten Suche nachvollziehbar. Die *Griff-Inspektion* zeigt bei Suchfehlern außerdem die verwendete ROI. Informationen zu erlaubten Kollisionen, nicht verwendeten Trajektorienschritten und Gründe für unerfüllte Entnahmebedingungen sind besser nachvollziehbar. Die Tasks *Griffe auslesen*, *Teil greifen* und *Griffsuchen-Statistik* bieten als optionale Ausgabevariablen zusätzliche Informationen zur Kollisionstiefe und zur Gelenkposition des Roboters am Griffpunkt aus.
- **Variablen:** Im *Variablen-Editor* werden Namen mit Zahlen in natürlicher Reihenfolge sortiert, etwa `Pose2` vor `Pose10`. Mit *Strg/Umschalt* können mehrere Variablen ausgewählt und mit *Entf* gemeinsam löschen. Einheiten und lokalisierte Zahlenformate machen Task-Werte leichter lesbar. Posen-Variablen können in weiteren Rotationskonventionen angezeigt werden.

- **IOs und JavaScript:** Die IO-Seite kennzeichnet die Ansteuerung über Roboter oder Feldbus. JavaScript-Aufrufe für IOs akzeptieren einen expliziten Namensraum. Hinzu kommen Griffnamen eines Produkts und die aktuelle Pose des gegriffenen Teils.
- **Bezeichnungen:** Die Bezeichnung `Experimentelle Griffe` wird durch `Hilfsgriffe` ersetzt. Für die bisher auch als Übergangspose bezeichnete Position wird einheitlich `Transitpose` verwendet.
- **Darstellung:** Die Mikado-Anwendung wird mit Bildschirm-Auflösungen ab 1280×1024 vollständig dargestellt. Ansichten können als gruppierte Tabs abgedockt und verschoben werden (ersetzt die bisherige Einstellung „separate Fenster“).

1.2 Version 3.4.0

Mikado Version 3.4.0 erweitert die Griffsuche, ergänzt neue Programmiermöglichkeiten und verbessert die Bedienung beim Starten, Sichern und Einrichten von Programmen.

Erweiterte Griffsuche

Die Tasks *Griffe auslesen*, *Teil greifen* und *Griffsuchen-Statistik* stellen zusätzliche Ausgabevariablen bereit. Dadurch können Programme direkt auf Informationen wie Fundanzahl, Anzahl greifbarer Funde, Fundpose, Griff, Greifer und Kollisionsart zugreifen.

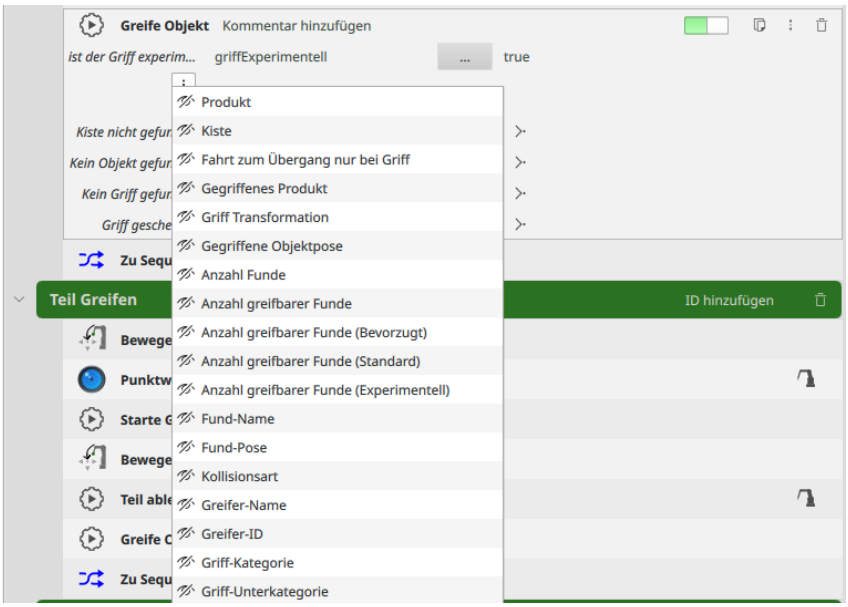

Abb. 8: Zusätzliche Ausgabevariablen im Task `Greife Objekt`.

Für die Griffsuche kann nun eine feste Z-Höhe relativ zur Kiste vorgegeben werden. Die kartesischen Bewegungen des Greifens und Entnehmens werden dadurch bis zu dieser Höhe geplant, was bei Gestänge, engen Kisten oder höheren Kanten die Pfadplanung besser kontrollierbar macht.

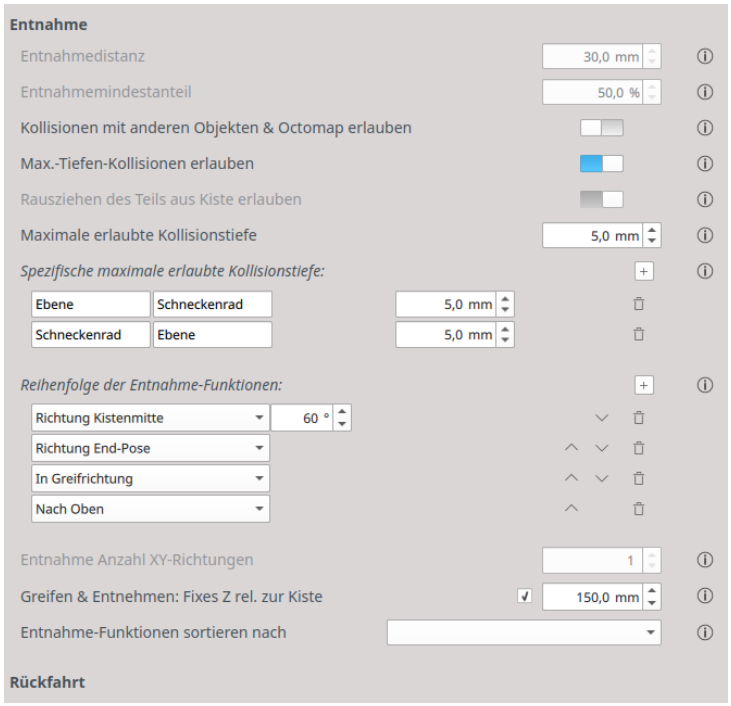

Abb. 9: Einstellung einer festen Z-Höhe relativ zur Kiste für Greif- und Entnahmepfade.

Außerdem können Grifftrajektorien geglättet werden. Dadurch entstehen weichere Übergänge zwischen Anfahrt, Griff, Entnahme und Rückzug.

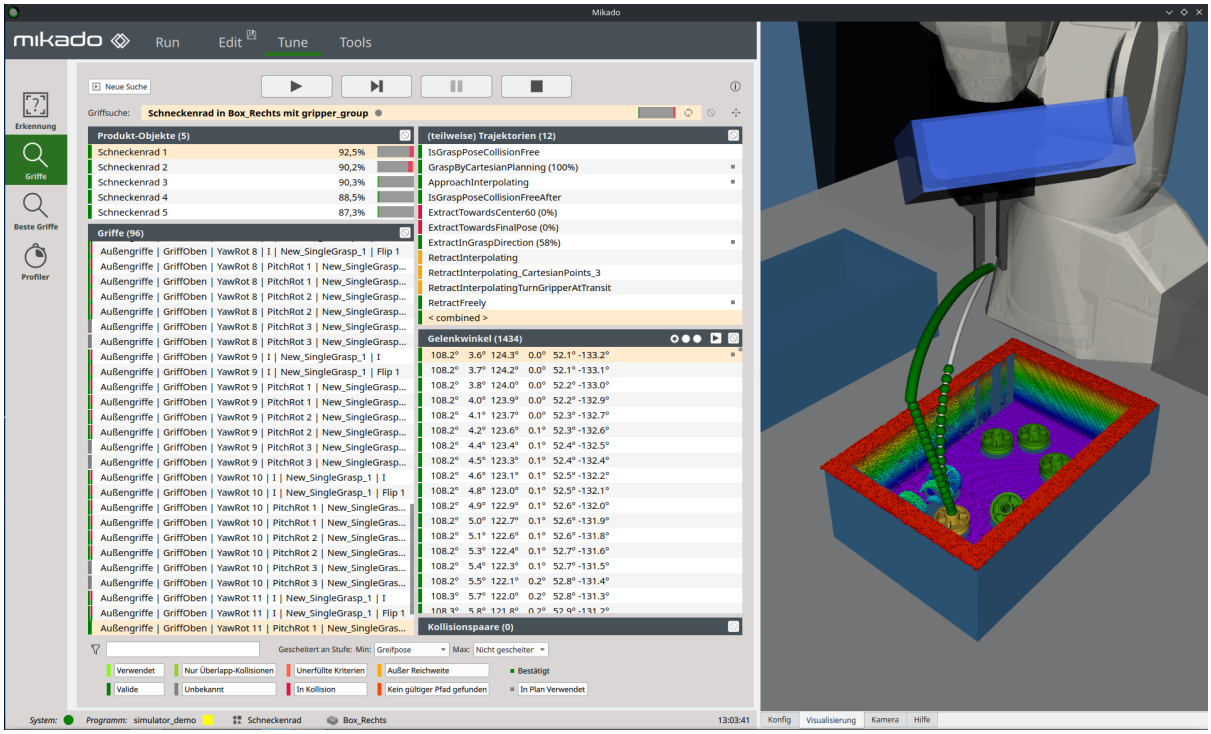

Abb. 10: Geglättete Grifftrajektorie in der Griffinspektion.

Neue Tasks und Schnittstellen

- **Roboterbewegungen:** Neue Tasks ermöglichen relative Bewegungen des Roboters, TCPs oder gegriffenen Teils.
- **Robotersignale:** Der neue Task *Auf Roboterausgangssignal warten* wartet auf Ausgangssignale des Roboters.
- **JavaScript:** Die Pose des gegriffenen Objekts kann über picksearch.getPickedObjectPose() abgefragt werden.
- **FANUC:** Die FANUC-spezifische Seite kann das Roboter-Webinterface anzeigen und im Browser öffnen.

Bessere Bedienung

- **Welcome-Seite:** Startdateien für Programme, Produkte und Szenenobjekte können direkt ausgewählt werden.
- **Systemseite:** Die aktuelle Zelle kann als ZIP-Archiv gesichert werden.
- **Programmeinstellungen:** Sprachauswahl, Fenstergröße, Fensterzustand und Vollbildmodus werden nach Änderungen im Mikado auf dem jeweiligen Controller gespeichert. Dadurch kann z.B. bei Offline-Inbetriebnahmen oder Fehlersuche bequem mit der eigenen Sprache gearbeitet werden, ohne die Zelle vom Kunden umstellen zu müssen.
- **Task-Editor:** Optionale Variablen ohne Wert werden zunächst ausgeblendet und können bei Bedarf über ein Menü hinzugefügt werden.

1.3 Version 3.3.0

Mikado Version 3.3.0 führt verschiedene neue Tasks, eine neue Feldbusseite sowie neue Fehlermeldungsfunktionalitäten ein und enthält zahlreiche Usability- und Stabilitätsverbesserungen.

Höhere Flexibilität bei Vakuumgreifern

Für Vakuumgreifer kann ab Mikado 3.3.0 ein Bereich definiert werden, der bei der Griffplanung nicht als Kollisionsobjekt berücksichtigt wird. Wie in dem unten gezeigten Beispiel, wird dabei eine Ebene relativ zum TCP definiert (mit einem festlegbaren Z-Offset). Diese Funktionalität erlaubt es unter anderem, flexible Saugnäpfe von Vakuumgreifern bei der Griffplanung von Kollisionschecks auszuschließen. Einstellbar ist die neue Funktion auf der Greiferseite des Mikado Assistant.

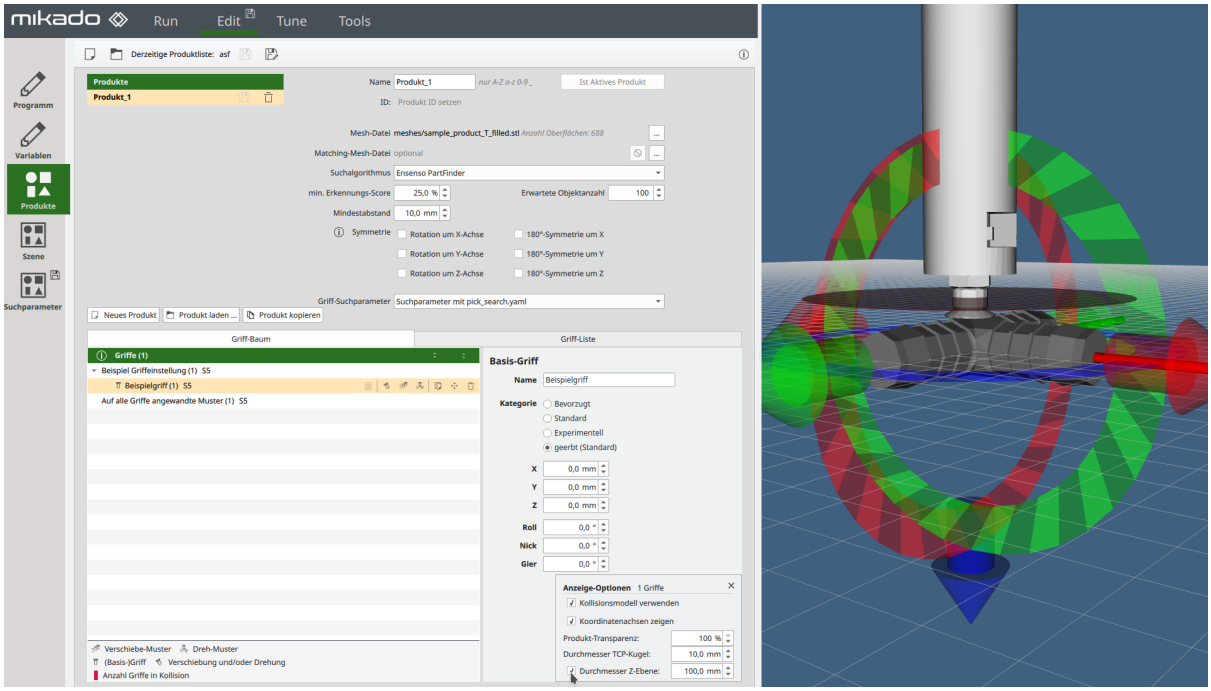

Abb. 11: Visualisierung des während der Griffsuche nicht kollisionsgeprüften Bereichs am Greifer. Alles unterhalb der dunkelgrauen Ebene wird bei der Griffsuche nicht als Kollisionsobjekt behandelt.

Darüber hinaus ist es nun bei verschiedenen Roboterherstellern möglich, den Vakuumgreifer nicht bereits an der Transitpose anzuschalten, sondern dies während der Fahrt bei Erreichen eines festlegbaren Abstands zu tun. Dadurch kann ungewolltes Ansaugen oder Verschieben von Teilen vor Erreichen der eigentlichen Griffpose minimiert werden.



Abb. 12: Einstellungsmöglichkeit der entfernungsbasierten Aktivierung eines Vakuumgreifers und des komprimierbaren Greiferbereichs im Mikado Assistant.

Zusätzliche Filtermöglichkeiten

Um bei komplexen Programmen die Handhabung zu erleichtern, wurden an verschiedenen Stellen Filtermöglichkeiten hinzugefügt. Durch Eingabe in die entsprechenden Textfelder können Variablen/Sequenzen einfacher gefunden werden.

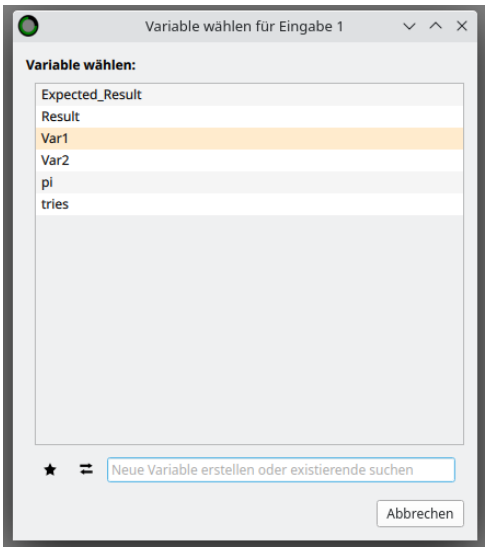

Abb. 13: Neuer Filter in der Variablenauswahl

Neue Filteroptionen gibt es an den folgenden Stellen:

- **Variablenauswahl:** Filtern der Kisten, ROIs, Produkte und anderer Variablen
- **Variableneditor:** Filtern nach Typ
- **Sequenzauswahl:** Filtern nach Namen
- **Programmeditor:** Filtern beim Springen zu Sequenzen
- **Beste Griffe:** Neuer Filter um ausschließlich Griffe anzuzeigen, die zum Zeitpunkt der Griffauswahl vorhanden waren
- **Griffinspektion:** Filtern nach Namen des Griffs

Neue und überarbeitete Tasks

In der Version 3.3.0 stehen drei neue Rechentasks zur Verfügung: *Arithmetische Operation*, *Logische Operation* und *Vergleichsoperation*. Der Task *Arithmetische Operation* ermöglicht die Durchführung einfacher Rechenoperationen mit zwei Variablen (Integer oder Double), mit Hilfe des Tasks *Logische Operation* können boolsche Verknüpfungen realisiert werden, und *Vergleichsoperation* ermöglicht das Vergleichen von zwei Variablen.



Abb. 14: Der neue *Arithmetische Operation* Task

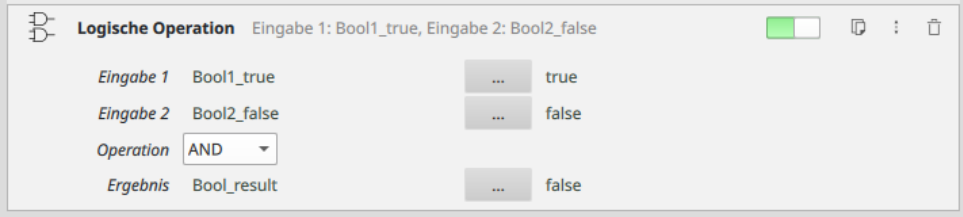

Abb. 15: Der neue *Logische Operation* Task

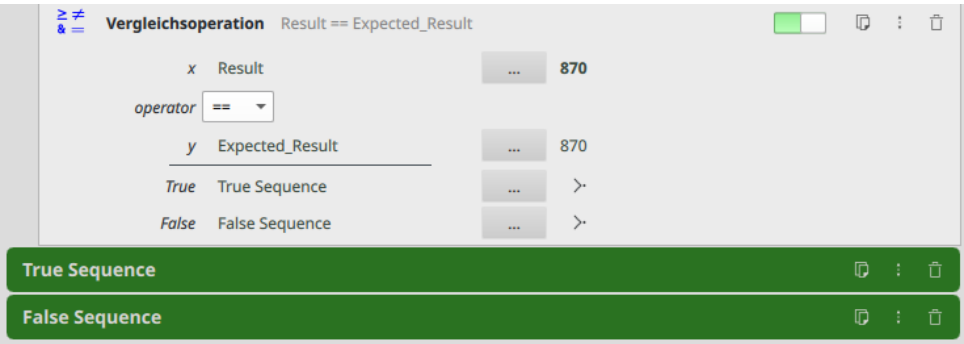

Abb. 16: Der neue *Vergleichsoperation* Task

Als weitere Neuerung steht jetzt der *Feldbusausgangssignal asynchron setzen* Task bereit, der einen Feldbusausgang setzt, sobald der Roboter eine festgelegte ROI verlassen hat. Das ermöglicht eine komfortable und schnelle Rückmeldung an die SPS, wenn der Roboter einen bestimmten Bereich der Anlage verlassen hat.

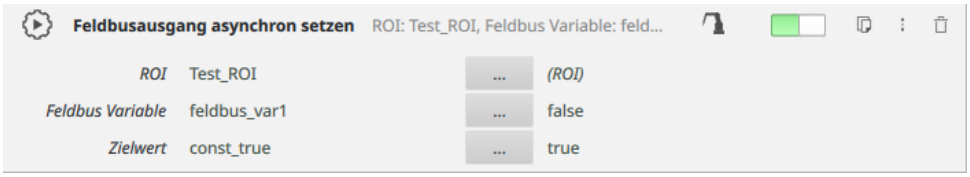

Abb. 17: Der neue *Feldbusausgangssignal asynchron setzen* Task

Weitere Änderungen von Tasks:

- *Pose erstellen* kann nun auch zum Überschreiben einer einzelnen Komponente (z.B. des Z-Werts) verwendet werden
- *Finde Objekt*, *Griffsuche starten* und *Griffe auslesen* nehmen keine Änderungen am aktiven Produkt mehr vor
- Im *Teil greifen* Task kann über einen optionalen Parameter eingestellt werden, dass nur bei einem gefundenen Griff die Fahrt zur Transitpose erfolgen soll

Erweiterte Funktionalität für JavaScript

Um komplexe Funktionalitäten noch besser umsetzen zu können, wurde der JavaScript-Task erweitert und folgende Verbesserungen hinzugefügt:

- Beim Umbenennen einer Mikado-Variable wird diese auch in *JavaScript* Tasks umbenannt
- XMLHttpRequests können optional erlaubt werden
- Bei Griffen aus der Griffsuche sind neben den TCP-Posen auch die Tool0-Posen zugänglich
- `lab.callService` erlaubt die Verwendung von beliebigen ROS Services
- native Roboterprogramme können aus JavaScript aufgerufen werden

Verbesserungen in der Griffsuche

Die Griffsuche wurde um verschiedene Parameter und Sortierkriterien erweitert. Außerdem wurden die Rückmeldungen an den Nutzer verbessert:

- *Griffsuche starten* akzeptiert nun einen Parameter für den (zukünftigen) Greifer
- **Positionsgleichheit**: Separate Parameter für Toleranz bei Wiedererkennung von erfolglos gegriffenen und wiedererkannten Objekten: `Teilepose Max. Translationsunterschied`, `Teilepose Max. Rotationsunterschied`, `Griff Max. Translationsunterschied` und `Griff Max. Rotationsunterschied`
- **Sortierkriterien:** Neue Sortierkriterien `Teile-Winkel zur Vertikalen` und `Anzahl Griffversuche`
- **Griffinspektion:** Neuer Button um Visualisierung auf Kistenmitte zu zentrieren
- **Griffinspektion:** Anzeigen wenn Transitpose nicht erreichbar ist oder andere Probleme bestehen
- **Griffsuche:** Verbesserte Berücksichtigung des User-Binnings

Verbesserte Nachvollziehbarkeit für Feldbus Kommandos

Um über einen Feldbus empfangene Kommandos besser nachvollziehen zu können, werden diese auf der neuen `Feldbus` Seite protokolliert. Diese kann in Mikado unter dem Menüpunkt `Run` gefunden werden und hält unter anderem Informationen über die Art der empfangenen Befehle, Startzeit und Dauer sowie den Befehlsstatus bereit.

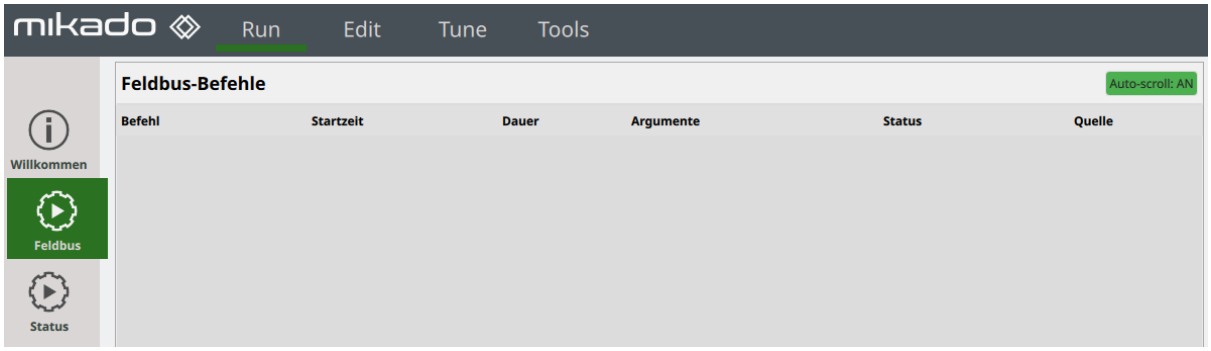

Abb. 18: Auf der neuen Feldbusseite werden alle eingehenden Kommandos erfasst und protokoliert.

Weitere Feldbus Neuerungen:

- **Feldbus Kommandos:** Neue Kommandos zum Starten und Auswerten einer Griffsuche direkt über den Feldbus
- **Mikado Status:** Verbesserte Statusmeldungen bezüglich des Ladens von Programmen, Produkten und Szenen

Objekterkennung

Mit der Version 3.3.0 wurde die Benennung der Symmetrien überarbeitet und weitere Symmetrien ergänzt. Alle Symmetrien werden bei der Objekterkennung und dem Task `Bewege Teil nach` berücksichtigt. Auf der Erkennungsseite kann im Modellgenerierungstab durch die neue Visualisierung der vom PartFinder berücksichtigten Ansichten je nach gewählter Symmetrie, die Auswirkungen der Symmetriewahl besser nachvollzogen werden.

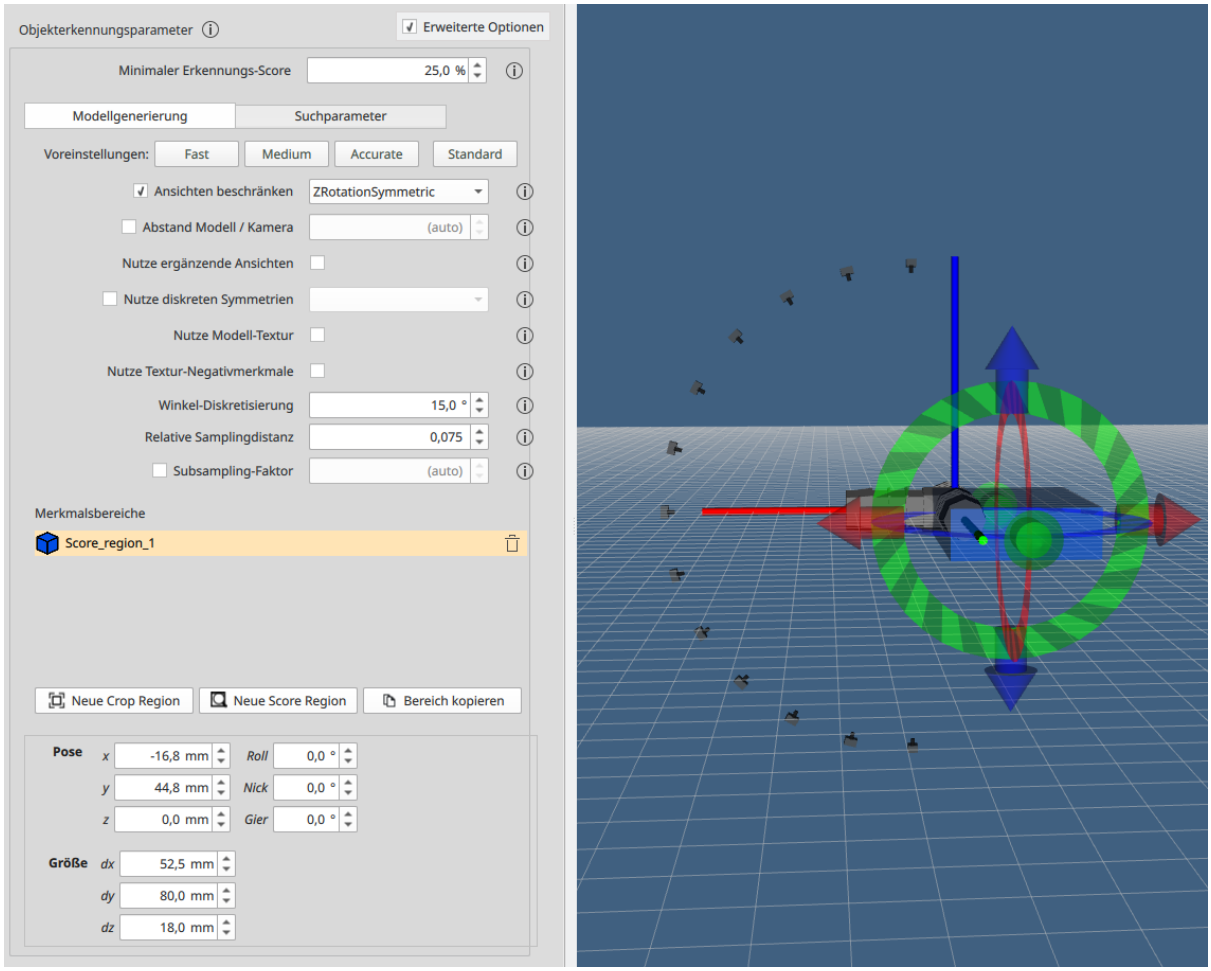

Abb. 19: Neue Visualisierung der Anzeige der bei der Objekterkennung berücksichtigten Ansichten und Beispiel einer `Score Region`

Außerdem wurden an Mikado Anpassungen vorgenommen, um die Ensenso SDK-Version 4.2 zu unterstützen, wodurch eine deutliche Verbesserung der Objekterkennung erreicht wird.

Zusätzlich wurden folgende Punkte verbessert:

- **Bild speichern:** Der aktuelle Produktname kann beim Speichern von Filecams als Prefix verwendet werden
- **Modellgenerierung:** Es können nun Bereiche des Modells definiert werden (`Score Region`), die priorisiert in die Berechnung des Erkennungs-Scores eingehen
- **Produkt Mesh:** OBJ-Meshes können jetzt direkt als Produkt Mesh ausgewählt werden
- **Hypothesen Clustering:** bis zu 4096 Cluster erlaubt (wie im Partfinder)

Debugging und Error Reporting

Um im Fehlerfall die Analyse zu vereinfachen, wurde auf der Seite Tools->Tools ein Button hinzugefügt, um einen Fehlerbericht zu erstellen. Nach einem Klick auf den Button werden Sie dazu aufgefordert, den beobachteten Fehler zu beschreiben. Nach einem Klick auf „OK“ wird automatisch eine Archivdatei mit relevanten Debugdateien erstellt, die die Untersuchung des Fehlers deutlich erleichtert.

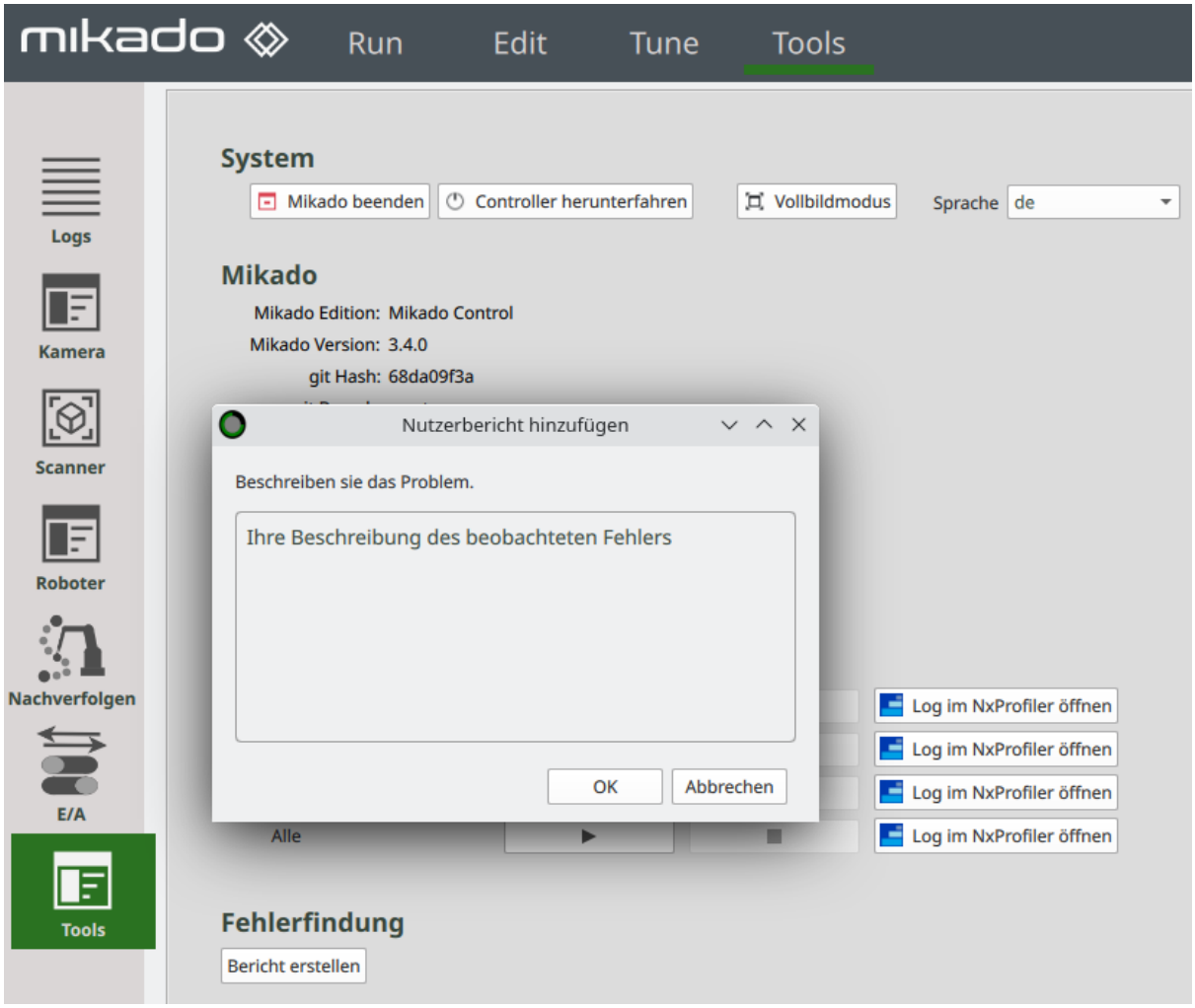

Abb. 20: Erstellung eines Fehlerberichts auf der Seite Tools->Tools

Die Log-Anzeige unter Tools->Logs wurde modernisiert. Die eigenständige Log-GUI kann nun auch gespeicherte Logs zur Analyse laden.

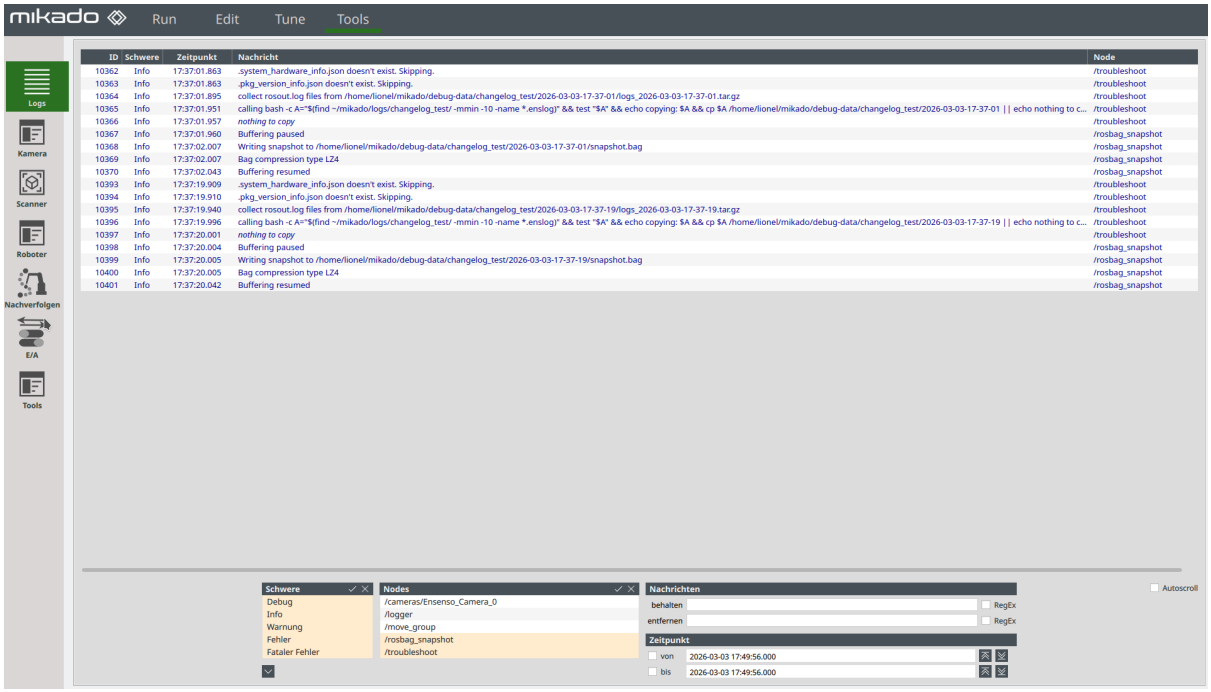

Abb. 21: Ansicht der modernisierten Seite Tools->Logs

Bessere Usability

- **Variablen:** Kameras und Greifer als Variablentyp
- **Griffeditor:** Vereinfachtes Einrichten durch optional transparente Produkte und an Produktgröße angepasste Marker
- **Kioskmodus:** Das Verlassen des Kioskmodus mit Alt+Tab etc. wird unterdrückt. Über eine KWin-Fensterregel kann auch die Windows-/Meta-Taste unterdrückt werden. Der Status und Logins im Kioskmodus werden geloggt.
- **Roboterseite:** Der aktive Wechselgreifer wird anzeigt und kann geändert werden
- **Inbetriebnahmemodus:** Neuer Einrichtmodus, in dem Handbetrieb nicht als Fehler im Roboterstatus kommuniziert wird
- **Szenenobjekte:** Kameraauswahl bei Aufnahmebutton

Verschiedenes

- **Protokollierung:** Im Ordner einer Mikado-Zelle werden nun neben den Softwareversionen auch die Hardware und IP-Adressen protokolliert
- **Roboterseite:** Anzeige von Informationen zum Robotermodell und Hersteller
- **Statusbar:** Techman-Fehlercodes werden passend zum Handbuch in Hex angezeigt
- **Suchstrategien:** Schrittweite der kartesischen Bewegungen ist einstellbar
- **Kuka:** Bei Robotern von Kuka sind Parameter zur Beeinflussung der Geschwindigkeit auf der Roboterseite sichtbar
- **MeshScanner:** Meshs nicht immer im OBJ-Format speichern
- **Mesh-Namen:** Mehr Sonderzeichen in Mesh-Dateinamen erlaubt
- **Programmeditor:** Warnung wenn Programm wegen voller Festplatte nicht geschrieben werden kann
- **Szenenobjekte:** Verbesserte Rückmeldung wenn Kistensuche scheitert

- **Optimierungen:** Optimierungen beim Laden und Validieren von Programmen sowie bei der Berechnung von Griffen
- **Bugfixes:** Diverse Bugfixes

1.4 Version 3.2.0

Griffsuche graphisch einstellbar

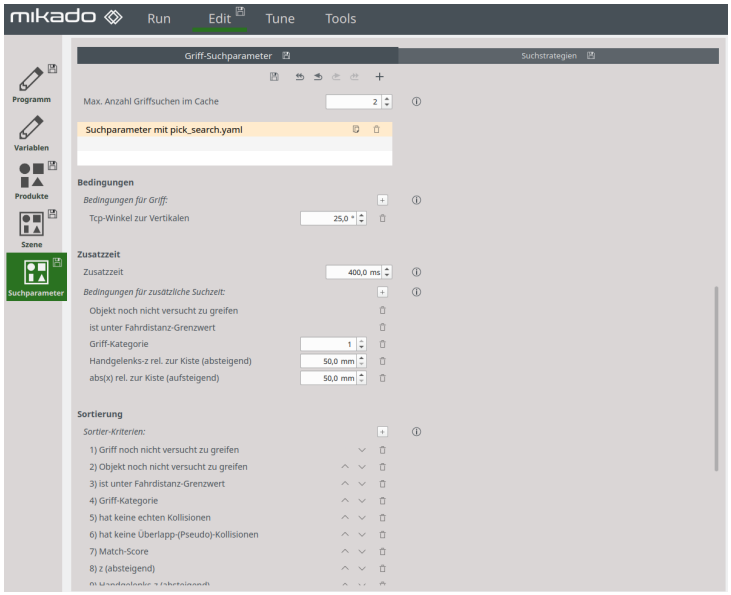

Abb. 22: Einstellung der Parameter einer Griffsuche

Die Parameter der Griffsuche können nun vollständig direkt in Mikado eingestellt werden und es können auch je nach Produkt oder Situation verschiedene Parametersätze verwendet werden. Damit kann beispielsweise bequem beeinflusst werden ob Teile strikt in einer gewissen Reihenfolge entnommen werden müssen oder ob beim Greifen bestimmte Strategien bevorzugt werden sollen. Die bisherigen, rudimentären Einstellmöglichkeiten im Mikado Assistant wurden mit dieser Version entfernt.

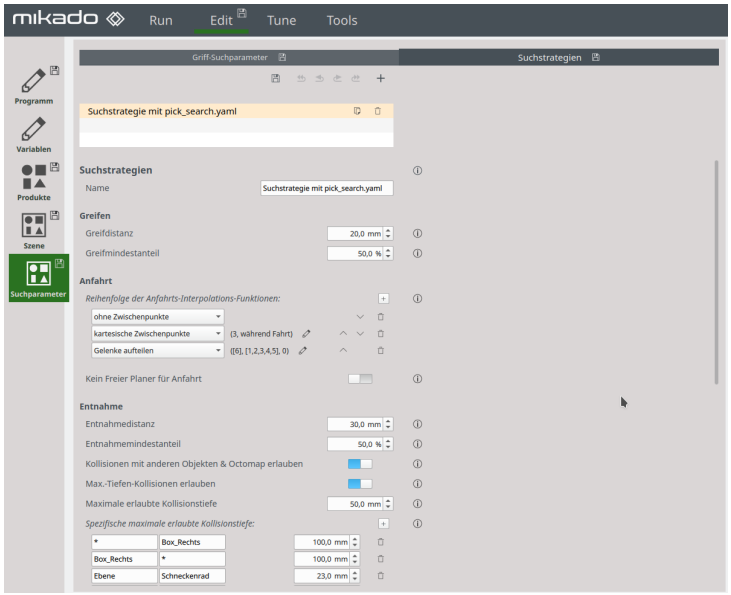

Abb. 23: Einstellung der Entnahmestrategien

Zusätzlich sind einige neue Extraktionsvarianten hinzu gekommen, insbesondere das seitliche Herausziehen im Objektkoordinatensystem, das sich für chaotisch geschichtete Blechteile bewährt hat.

Wechsel zwischen mehreren Griffsuchen

Es kann nun zwischen mehreren Griffsuchen gewechselt werden und zum Beispiel effizient abwechselnd aus mehreren Kisten gegriffen werden. Die Griffsuchen können dabei alle in der Griffinspektion bzw. in der Liste der besten Griffe untersucht und optimiert werden. Es können auch verschiedene Greifer durchgerechnet werden, ohne eine neue Objekterkennung starten zu müssen. Dies erfolgt automatisch bei einem Starte Griffsuche ohne neue Aufnahme oder über JavaScript (siehe *rerunPickSearch*).

Numerische IDs für Greifer, Kameras, Produkte und Szenenobjekte

Für eine vereinfachte Steuerung des Prozessablaufs über eine SPS können nun numerische IDs zusätzlich zu den Namen verwendet werden, um Greifer, Kameras, Produkte oder Szenenobjekte zu wechseln.

Gelenkwinkelgrenzen besser verständlich

Es wird nun klarer kommuniziert, wenn der Roboter sich außerhalb der Gelenkgrenzen befinden. Sei es beim Programmstart, beim Jogging oder bei Gelenkpositions-Variablen im Editor.

Neue PartFinder Features

Es wurden einige neue Parameter der PartFinder Objekterkennung des Ensenso SDKs 4.1 zugänglich gemacht, dazu zählen die TexturePenalty oder das ComplementViewpoints. Weiterhin ist es nun möglich die Objekterkennung auf Ausschnitte des Models zu begrenzen ohne dazu das CAD Modell künstlerisch bearbeiten zu müssen. Dies erfolgt über sogenannte Merkmalsbereiche, die in der Benutzeroberfläche visualisiert werden.

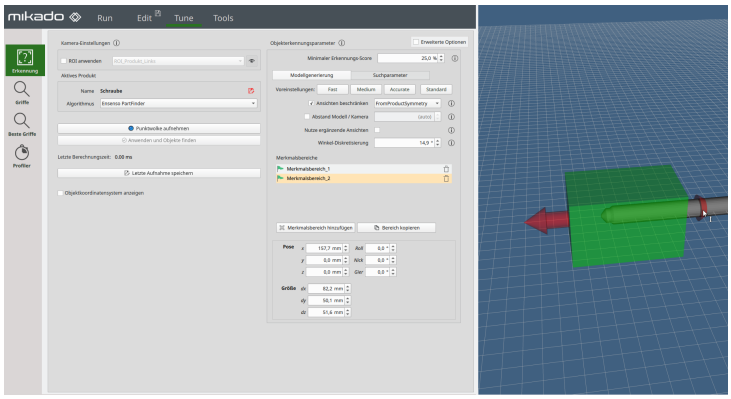

Abb. 24: Beispiel für Merkmalsbereiche: Nur die Spitze einer Schraube soll beachtet werden

Für eine vereinfachte Interpretation der Objekterkennungsergebnisse ist es nun möglich, die Koordinatensysteme der Funde einzublenden.

Griffe auslesen in Mikado Select

Die kollisionsgeprüfte Griffsuche in Mikado Select wird nun wie im Mikado Control über Starte Griffsuche und den neuen Task Griffe auslesen asynchron durchgeführt. Der bisherige Finde Objekte Task wird automatisch beim Upgrade durch die beiden Tasks ersetzt wenn die Griffsuche bisher verwendet wurde.

Kleinere Verbesserungen

- Programm: Tasks können deaktiviert bzw auskommentiert werden und werden dann übersprungen
- Jogging: Suchen nach Variablenname
- Filtern von Variablen im Auswahldialog
- JS: IOs lesen und schreiben (siehe *robot*)

- JS: Slot um Task auf ErrorMightRetry zu setzen (siehe *task*)
- JS: lab.callExtensionService für Erweiterungen (siehe *callExtensionService*)
- WaitForFieldbusInputSignalTask: Task zum Warten auf eine Wertänderung in einer Feldbusvariable (siehe *Auf Feldbuseingangssignal warten*)
- SetInitialValueTask: optionales Speichern des Programms (siehe *Ausgangswert setzen*)
- Auch Variablen von „Addons“ anzeigen
- Status: Springen zum aktiven Task
- Yaskawa-spezifische Funktionen und Fehlermeldungen
- Beim FindObjectTask wird eine ggf. parallel laufende Griffsuche nicht mehr unnötig beendet
- Anzahl Kontaktpunkte bei Kollisionsprüfung einstellbar
- Anzahl Kontaktpunkte bei Kollisionsprüfung erhöht für genauere Eindringtiefenberechnung
- PickInspektor: Bei herausgefilterten Griffen Begründung detailierter
- PickInspektor: Bessere Erläuterungen wenn Griff per Definition mit Objekt kollidiert
- Verwendung von Variablen in JavaScript Tasks auf Variablenseite anzeigen
- Anzahl Mesh-Dreiecke in Szenenobjekten anzeigen
- Bei Bewegungsfehler asynchrone Aufnahme sofort abbrechen
- Neues Profiler-Format verwenden
- separates Paket mikado-robots-common für Helferprogramme
- Usability Verbesserungen am Griffeditor
- Filecams werden unter mikado/filecams gespeichert und die Größe limitiert
- Neuer Task: Szenenobjekt-Position zurücksetzen

1.5 Version 3.1.2

Referenzen auf Szenenobjekte und Produkte

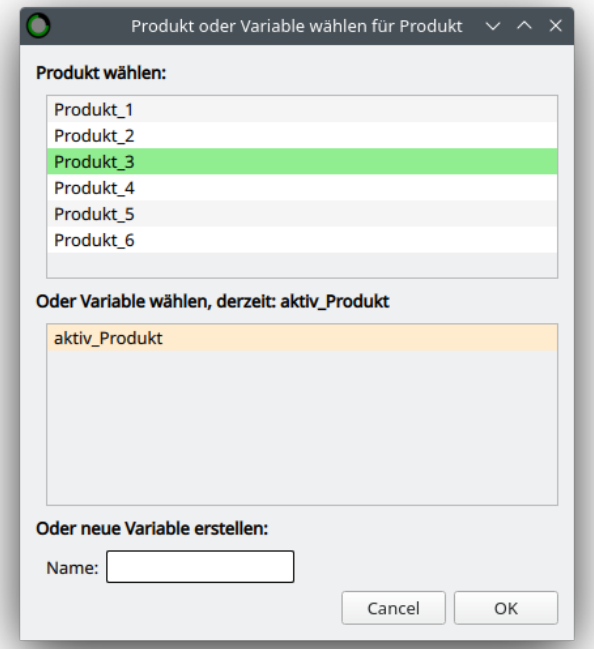

Abb. 25: Auswahl einer Programm-Variable oder einer Produkt-Referenz

Produkte und Szenenobjekte können in Tasks direkt referenziert werden. Dieses Feature kommt insbesondere dann zur Geltung, wenn innerhalb eines Tasks stets die selbe ROI genutzt oder das selbe Produkt gesucht werden soll. Das Setzen des Wertes via Programm-Variable ist weiterhin möglich.

Überarbeitung Roboter Nachverfolgen

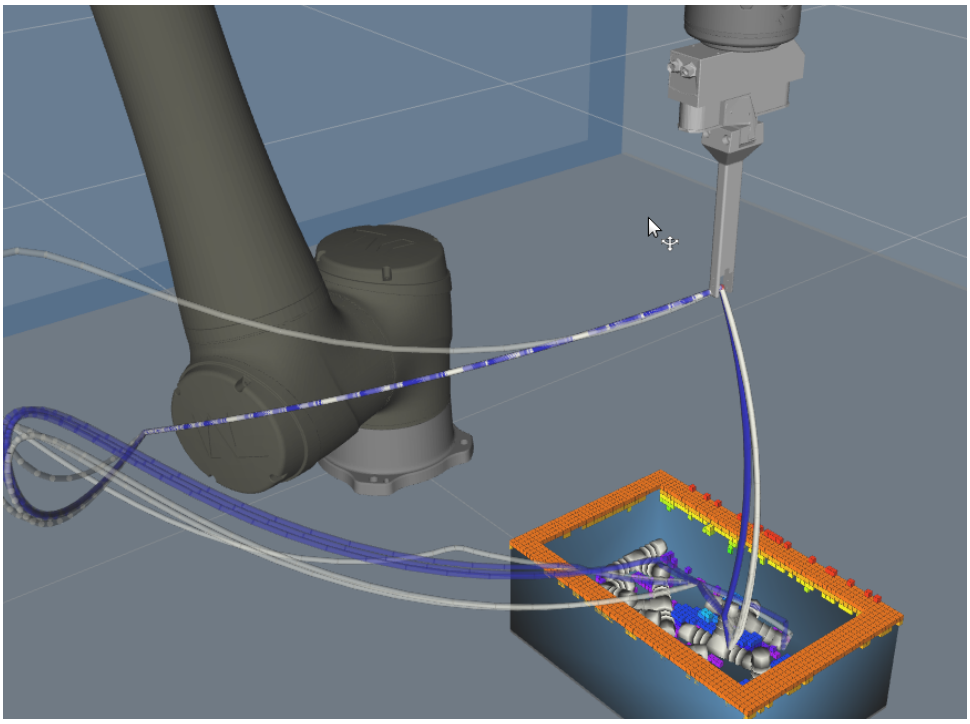

Abb. 26: Visualisierte Bahn des aktiven TCPs

Auf der *Roboter Nachverfolgen*-Seite wird die Bahn des aktiven ToolCenterPoints (TCP) als Linie visualisiert. Des Weiteren unterstützt die Seite die Verwendung von Wechselgreifern.

Greifer mit Positionskontrolle

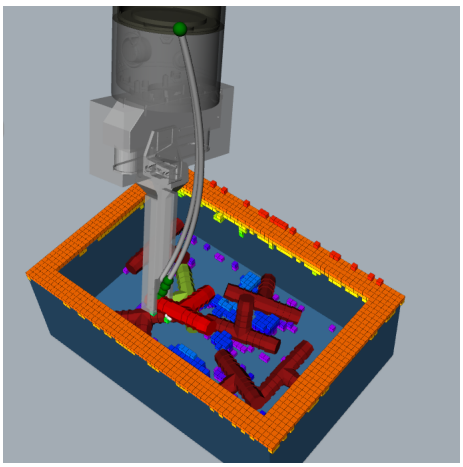

Abb. 27: Griff mit geringer Öffnungsweite

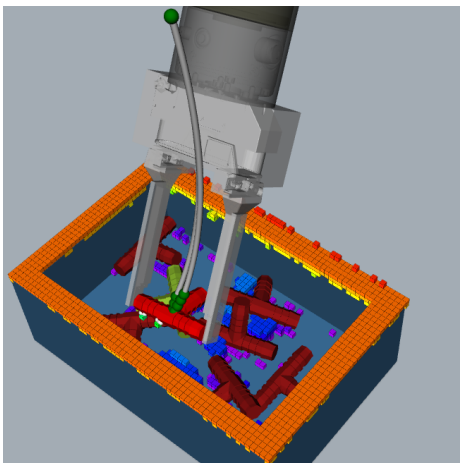

Abb. 28: Griff mit hoher Öffnungsweite

Für Greifer mit Positionskontrolle lässt sich die Öffnungsweite der Greiferfinger für jede Griff-Kategorie einstellen. Hierfür muss in der entsprechenden `gripper.yaml` der Typ des Greifers auf `type: ParallelPositionControl` gesetzt sein.

Mesh-Scanner

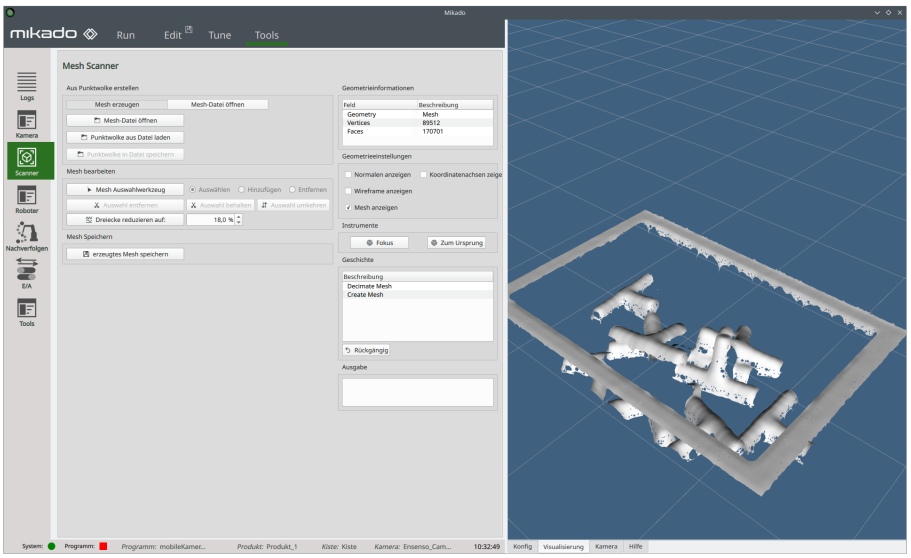

Abb. 29: Screenshot der Scanner-Seite

Die Scanner-Seite bietet die Möglichkeit ein Mesh aus einer Punktwolke zu erstellen und das erstellte Mesh nachzubereiten. Das Mesh kann anschließend in der Produktsuche verwendet werden. Der Scanner kann:

- Punktwolke erfassen oder Punktwolke/Mesh laden
- Mesh aus Punktwolken erstellen
- Mesh vereinfachen
- Teile des Meshes ausschneiden
- Mesh speichern

Kollisionsprüfung der Greifposen im Griff-Editor

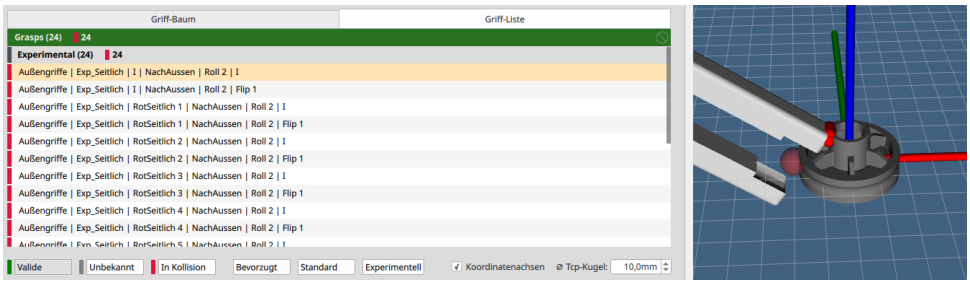

Abb. 30: Griff-Liste und Visualisierung der Kollisionspunkte

Neu erstellte Griffe werden beim Anlegen auf Kollision mit dem Teil geprüft (siehe *Kollisionsprüfung der Greifposen*). Neben dem Griff-Baum gibt es den Reiter Griff-Liste. Diese Liste enthält alle durch den Griff-Baum generierten Griffe. Wählt man in der Griff-Liste einen Griff aus, wird dieser in der Visualisierung angezeigt. Bei einem kollidierenden Griff werden die erkannten Kollisionspunkte zwischen Greifer und Produkt durch kleine rote Kugeln markiert.

Kleinere Optimierungen

- Deutlich schnellere Profinet-Schnittstelle durch event-basiertes Backend. Details siehe: *Version 3.1.0*
- GUI-Sprache auch ohne Mikado Neustart änderbar
- Roboterstatusanzeige aufgeräumt, mehr Fehlerdetails
- Status und Programmeditor: informativere Platzhalter-Texte
- Update Octomap jetzt auch aus JS möglich, ohne Griffsuche / FindeObjekte (siehe *updateOctomap*)
- Griff-Inspektor: Griffsuche kann manuell gestartet werden
- Griffe nach *Kollisionstiefe* sortierbar
- JavaScript: CSV Datei in Dict einlesen, zB für Produktparameter (siehe *file*)
- Warnung falls ein Produkt sehr große Meshes verwendet
- Fehlerhafte Variablen werden auf Variablenseite angezeigt, z.B. Schreiben in konstante Variable oder Variablen, die bei der Verwendung einen falschen Typ haben
- JavaScript: Tcp Koordinaten für Start, Approach, Retract, Final für Griffe (siehe *picksearch*)
- QML: es ist jetzt möglich beliebige Prozesse zu starten, zB Okular für eine projektspezifische Anleitung
- Schnittstelle um die Objekterkennung mit einem projektspezifischen Algorithmus umzusetzen
- PickSearch/JS: Zusätzliches frei nutzbares Bewertungskriterium für Objekte (User Binning)
- Position eines gegriffenen Teils anpassen
- Bewege Roboter nach: eventuelle Kollisionen direkt im Status des Tasks
- Griff-Editor: Kopieren und Verschieben von Griffen
- X > Y? und ähnliche Tasks mit blauen Symbolen für Logik/Sprünge
- Task um die Länge einer Liste auszulesen
- FindeObjekte: optionaler Sequenzwechsel falls Kiste gesucht und nicht gefunden wurde
- JS: Mehr Informationen über den letzten Griff
- Griffe nach Kollisionstiefe sortierbar

1.6 Version 3.0.5

- Fehlerkorrekturen zur Behandlung neu erstellter Variablen

1.7 Version 3.0.4

- Cuda kann für die Produktsuche via PartFinder aktiviert werden
- Sehr kurze kartesischen Fahrten werden nicht mehr auf 10 Punkte erweitert

1.8 Version 3.0.3

- Für Entwickler ist „Finde Objekte“ auch in Mikado Control sichtbar
- Finde Objekte hat nun auch einen optionalen Parameter „Box“
- PartFinder: maximale Anzahl der Cluster auf 200 erhöht
- neuer Task: Tool Center Point (TCP) kartesisch bewegen, optional mit Toleranz zum Ziel.
- Informationen über die aktuell verwendete FileCam anzeigen auf der Objekterkennungsseite
- GriffEditor: Verschiebungen können nun auch symmetrisch angelegt werden

1.9 Version 3.0.2

- Griffe: Gesamte Trajektorie in Vorschau visualisierbar
- „read-only“ Variablen
- Variablenseite: Favorisierte Variablen

1.10 Version 3.0.1

- Detaillierte Anzeige der Objektsuche / Griffsuche Pipeline
- ADS Schnittstellen-Support
- NxLogs Gui für Profiling

1.11 Version 3.0.0

Kollisionsprüfung des Greifers für Mikado Select

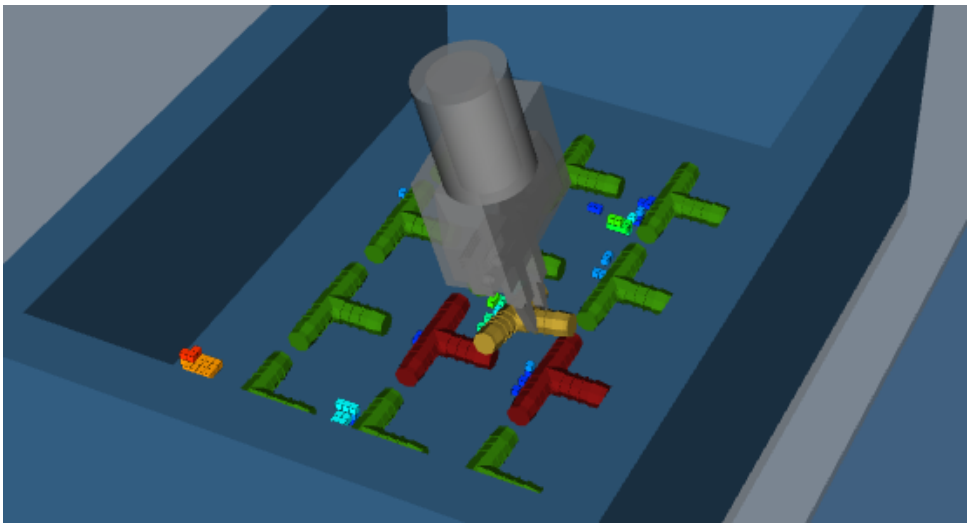

Abb. 31: Kollisionsprüfung des Greifers für Mikado Select

Mikado Select unterstützt das Finden von kollisions-geprüften Griffen. Der Task *Finde Objekt* besitzt hierfür einen neuen Rückgabewert `Valide Griffe`. Diese Posen-Liste enthält den besten Griff pro gefundenem und greifbarem Objekt. Die Liste ist sortiert nach benutzerspezifischen Kriterien (siehe *Beste Griffe*).

Visualisierung der Griff-Priorisierung

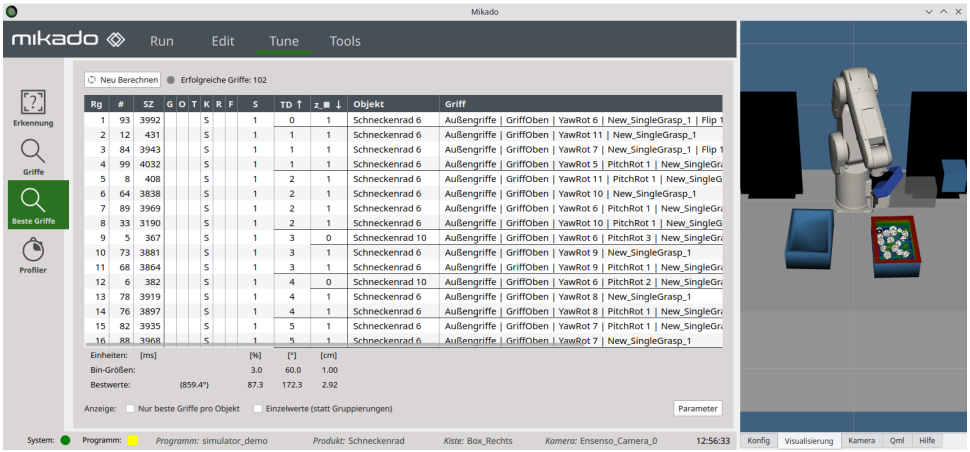

Abb. 32: Screenshot Beste-Griffe

Sortier- und Ausschluss-Kriterien für Griffe können global (für alle Produkte) in der `pick_search.yaml` definiert werden. Um alle Griffe bezüglich der verschiedenen Kriterien vergleichen zu können wird Binning verwendet. Die neue Seite *Beste Griffe* bietet ein Analyse-Tool für die Sortier-Kriterien.

Unterstützung für Wechselgreifer

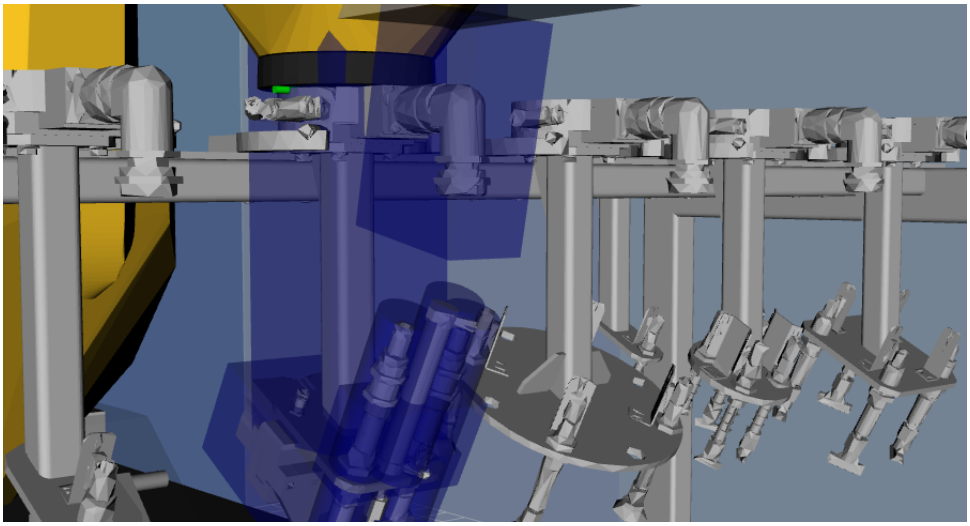

Abb. 33: Unterstützung für Wechselgreifer

Mikado Control unterstützt das Wechseln von Greifern zur Laufzeit einer Roboter-Zelle. Mit dem neuen Task *Setze aktiven Greifer* kann innerhalb des Roboterprogramms zwischen den definierten Greifer-Geometrien gewechselt werden. Produkte haben nun Greifer-abhängige Griffdefinitionen, siehe *Einrichten von Griffen bei Wechselgreifern*. Die Griffsuche prüft jeweils nur Griffe die für den aktiv gesetzten Greifer definiert wurden.

Das Einrichten einer Zelle mit mehreren Greifern wird vorerst vom Mikado-Support übernommen. Das Hinzufügen von neuen Greifern in Mikado Assistant ist in Arbeit und für das nächste Release angestrebt. Das Editieren von bereits eingerichteten Zellen in Mikado Assistant wird voll unterstützt.

Zusatzgelenke auch bei Parallelgreifern

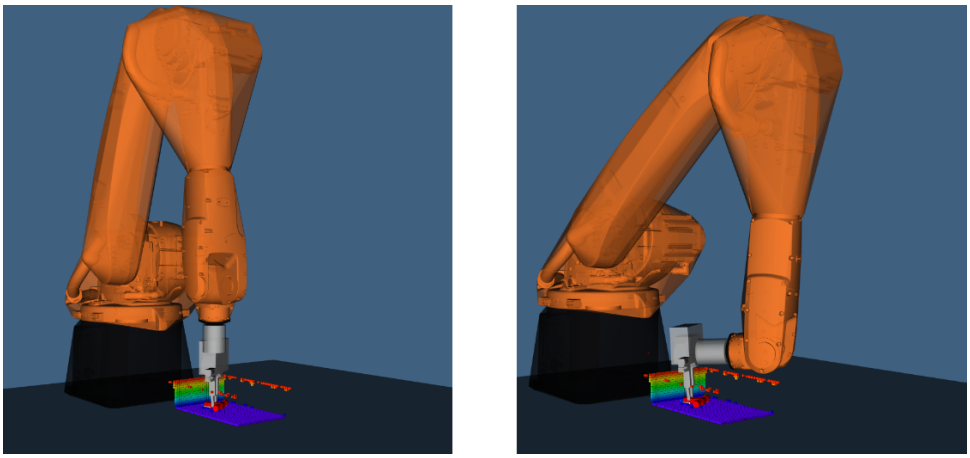

Abb. 34: Kippgelenk für Parallelgreifer

Zusätzliche Gelenke am Greifer werden auch für Parallelgreifer unterstützt. Zuvor waren nur Magnet- und Vakuumgreifer möglich (Geometrie ändert sich nicht beim Öffnen/Schließen). Weitere Details sind in der Dokumentation des *Greiferzustand einstellen*-Task zu finden.

Bibliothek mit Beispielprogrammen

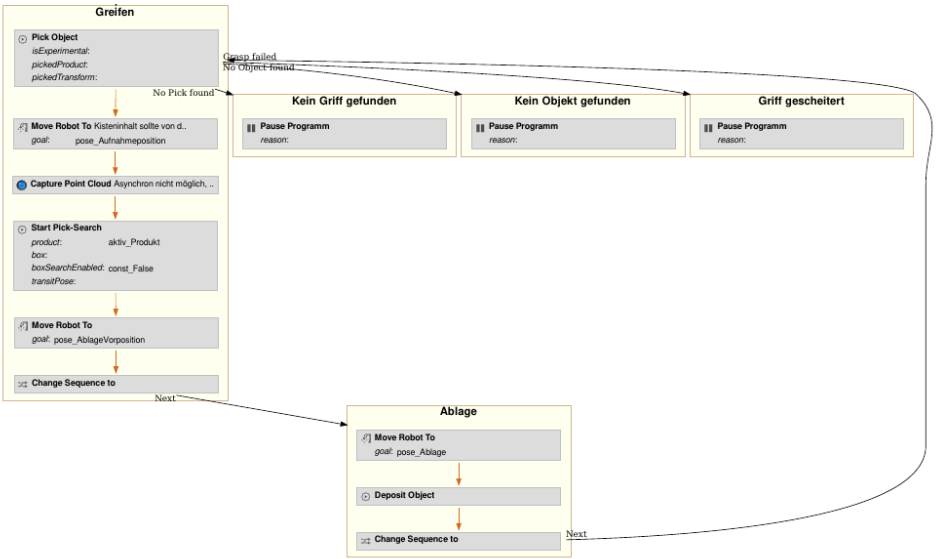

Abb. 35: Beispielprogramm für mobile Kamera

Die Programm-Bibliothek befindet sich auf der Willkommen-Seite und lässt sich mit einem Klick auf das Ordner-Symbol links neben der Überschrift ‚Programme‘ öffnen. Sie bietet Beispielprogramme die als Ausgangspunkt für Projekt-spezifische Programme geeignet sind. Darunter sind:

- **Select**
  - TCP-Kalibrierung Programm
  - Objekte finden und Übertragen der Ergebnisse via Profinet-Schnittstelle
- **Control**
  - TCP-Kalibrierung Programm

– Asynchrone Bildaufnahme mit stationärer Kamera
– Synchrone Bildaufnahme mit mobiler Kamera

If-Else Tasks für bedingtes Ausführen von Teilsequenzen

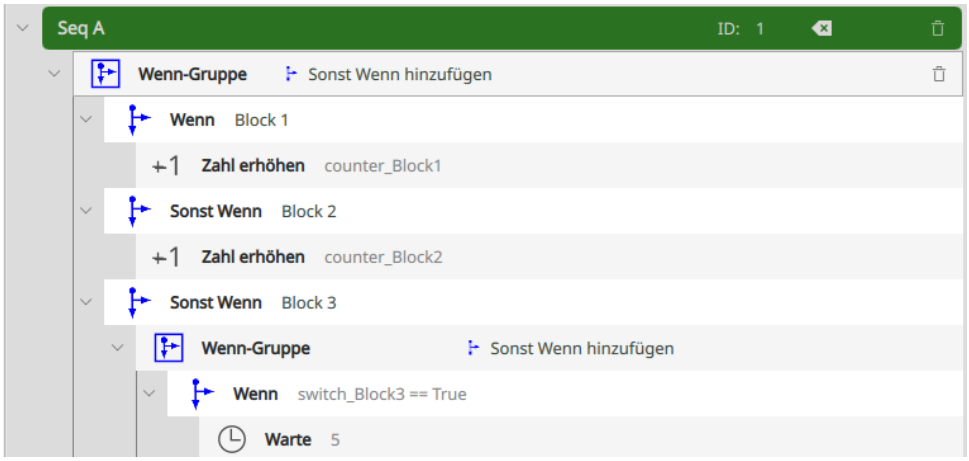

Abb. 36: Verschachtelte Wenn-Gruppen im Programm-Editor

Der neue *Wenn-Gruppe*-Task ermöglicht das bedingte Ausführen von Teilsequenzen. Dieses Feature ermöglicht unter Anderem auch:

- Das Kopieren von Sequenzen
- Das Untergliedern von Sequenzen in Teilsequenzen
- Das parallele Löschen von mehreren Tasks
- Re-Design des Programm-Editors

Rotation auf Jogging Seite

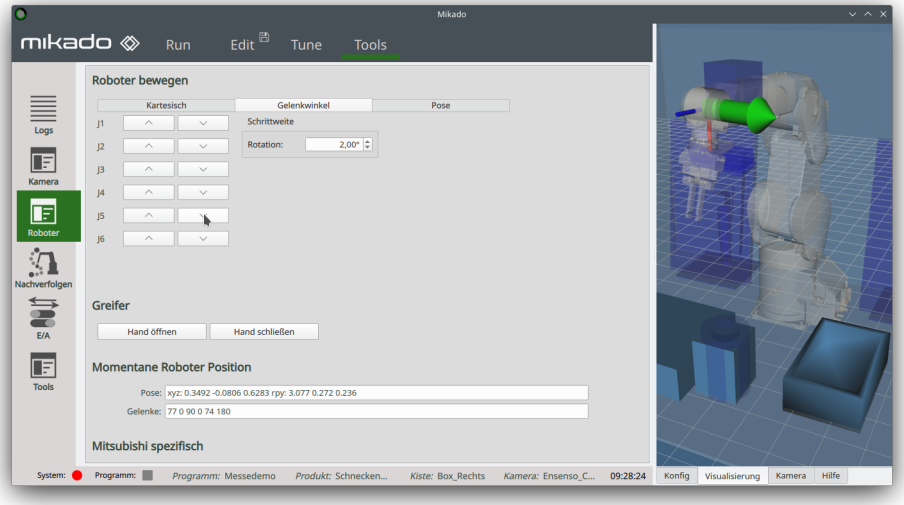

Abb. 37: Jogging Seite: Vorschau der Drehachse durch grünen Pfeil

Zusätzlich zu kartesischen Fahrten in x-, y- und z-Richtung, lässt sich der Roboter auf der Roboter-Seite (siehe *Robotersteuerung*) stückweise rotieren. Dabei kann manuell ausgewählt werden in welchem Koordinatensystem

rotiert wird. Zu Verfügung stehen hierbei folgende Koordinatensysteme:

- Welt - Koordinatensystem
- Roboter - Koordinatensystem
- Tool0 (bzw. Roboter-Handgelenk) - Koordinatensystem

Weitere Neuerungen:

- Die Bewegung wird auf der rechten Seite als Vorschau angezeigt
- Re-Design der Roboter-Seite

Depalettierungs-Regeln

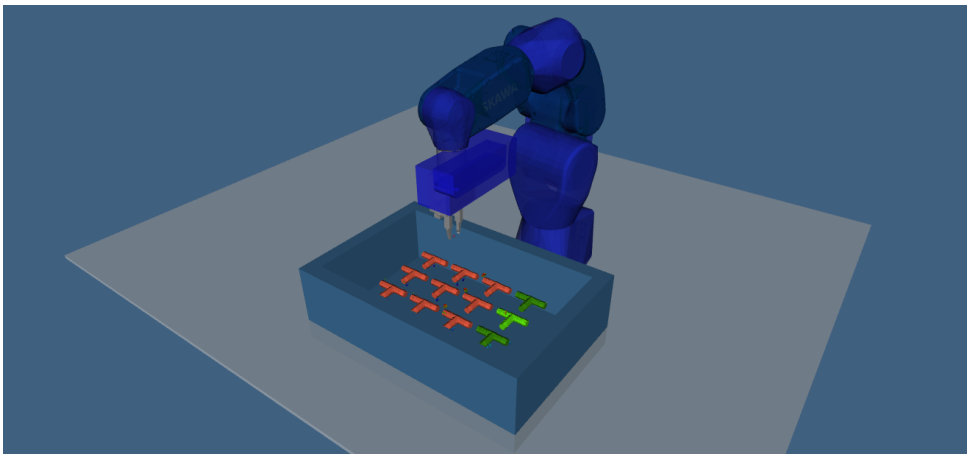

Abb. 38: Nur Griffe mit kleinstem x-Wert sind erlaubt

Der neue Parameter `max_bins_required` ermöglicht das Ausschließen von bestimmte Funden (siehe *Bedingungen*). So können beispielsweise Produkte mit hohem x-Wert relativ zu anderen Produkten ausgeschlossen werden. Dies ermöglicht das Depalettieren von links nach rechts.

Mehr Informationen zu gescheiterten Griffen

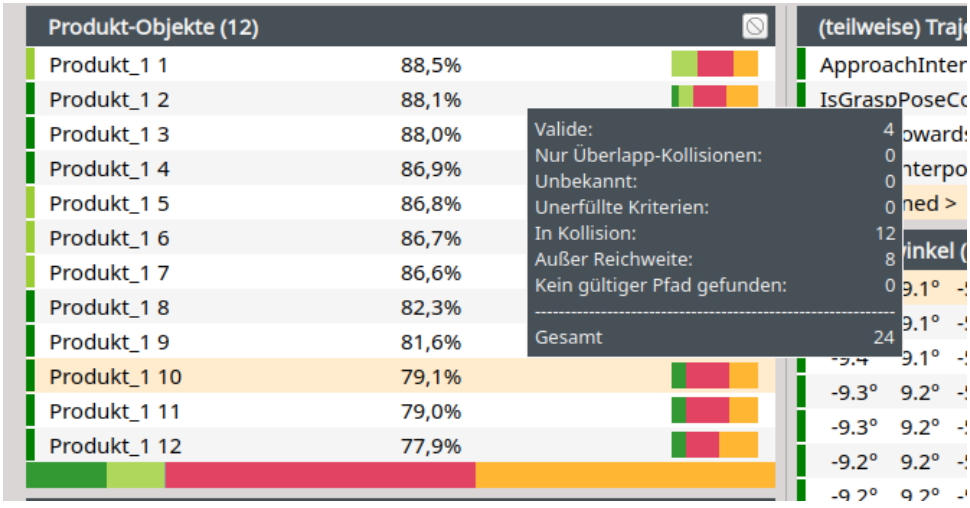

Abb. 39: Re-Design der Griffe-Seite

Die Griffe Seite wurde überarbeitet und bietet jetzt folgende zusätzliche Feature:

- Kollisionspaare haben teilweise zusätzlichen Erklärungstext
- Balken-Diagramme zeigen aktuellen den Zustand der Griffsuche insgesamt und den Produkt-spezifischen Zustand

JavaScript: API für Szenenobjekte

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

Abb. 40: Setzen des z-Wertes einer ROI auf einen fixen Wert

Über die JavaScript-API können nun Größe und Position von Szenen-Objekten gelesen und geschrieben werden (siehe *scene*). Folgende neue Funktionen stehen zur Verfügung:

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

1.12 Version 2.1.0

- Integration der Ensenso PartFinder Objekterkennung
- Objekterkennungsparameter werden produktspezifisch abgespeichert und eingestellt
- Gesamtanzahl Griffe wird beim Produkt angezeigt
- Es können sowohl einzelne Griffmuster als auch alle resultierenden Griffe angezeigt werden

Griffinspektion:

- Objekte werden je nach Greifbarkeit eingefärbt und können direkt angeklickt werden
- Griffe können nach Erfolg gefiltert werden

Erweiterungen:

- Die Ergebnisse der Griffsuche können nun per JavaScript abgerufen werden
- JavaScript Kinematik API um zB die Erreichbarkeit von mehreren Posen oder Pfaden zu prüfen
- Die Dimensionen der Punktwolke innerhalb einer ROI kann per JavaScript abgefragt werden
- Kalibrierungen über eine ROS-Schnittstelle (und damit zB über OpcUA oder ProfiNet)

2 Szenen-Objekte

2.1 Version 5

Warnung

Wenn Ihr Programm einen *Szenenobjekt-Position auslesen* oder *Szenenobjekt-Position setzen* verwendet, kann sich das Verhalten des Programms ändern. Bitte überprüfen Sie Ihr Programm entsprechend.

Die Bedeutung der `Default-Pose` eines Szenenobjekts wurde geändert. In Version kleiner 5 beschreibt die Default-Pose den Mittelpunkt des Bodens eines Szenenobjekt. Für die Ermittlung des Bodens wurde der Parameter `Default-Größe` verwendet. Somit war die tatsächliche Pose des Szenenobjekts abhängig von der Default-Pose

und der Default-Größe. Die Abhängigkeit zur Default-Größe wurde entfernt. Ab Version 5 beschreibt die Default-Pose bei Szenenobjekten mit Quader-Geometrie (wie bspw. ROIs) die Lage des Volumenschwerpunkts, statt wie zuvor den Mittelpunkt des Bodens. Bei Szenenobjekten mit Mesh-Geometrie (wie bspw. Kisten) beschreibt die Default-Pose nun direkt die Lage des Ursprungs eines Meshes im Welt-Koordinatensystem. Zusätzlich wurde der neue Parameter `Mesh-Offset` eingeführt, um den Ursprung eines Meshes aus Mikado verschieben zu können.

Mit dem Version-Update werden die verschiedenen Posen-Parameter für Szenen-Objekte automatisch auf die neue Bedeutung umgestellt. Folgende Operation werden durchgeführt:

1. Verschieben der `Default-Pose` um die Hälfte der z-Komponente der Default-Größe d_z in positive z-Richtung des lokalen Koordinatensystems,
2. Verschieben der `Übergangspose` um den gleichen Betrag und in entgegengesetzte Richtung und
3. Setzen der z-Komponente des `Mesh-Offset` auf $-\frac{1}{2}d_z$ (gleicher Betrag wie bei 1 und 2).

Die neue Bedeutung der Default-Pose ist in nachfolgender Abbildung dargestellt.

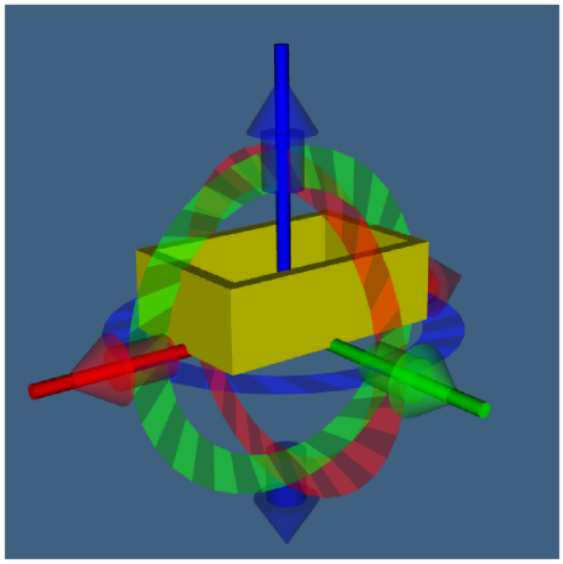

Abb. 41: Alte Bedeutung der `Default-Pose`

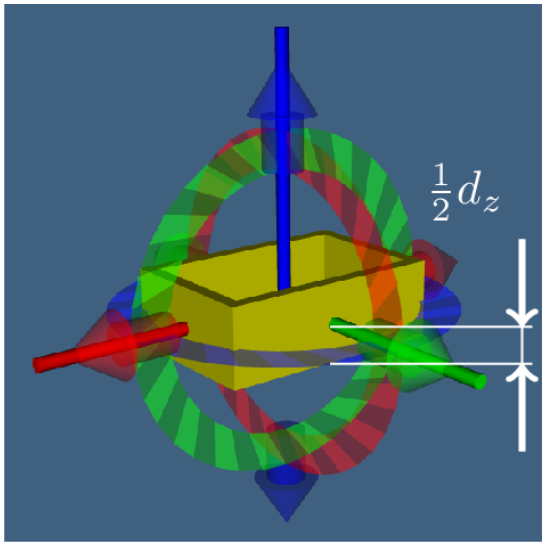

Abb. 42: Neue Bedeutung der `Default-Pose`

Die Default-Größe wird wie in folgenden Screenshots gezeigt in Mikado eingestellt.

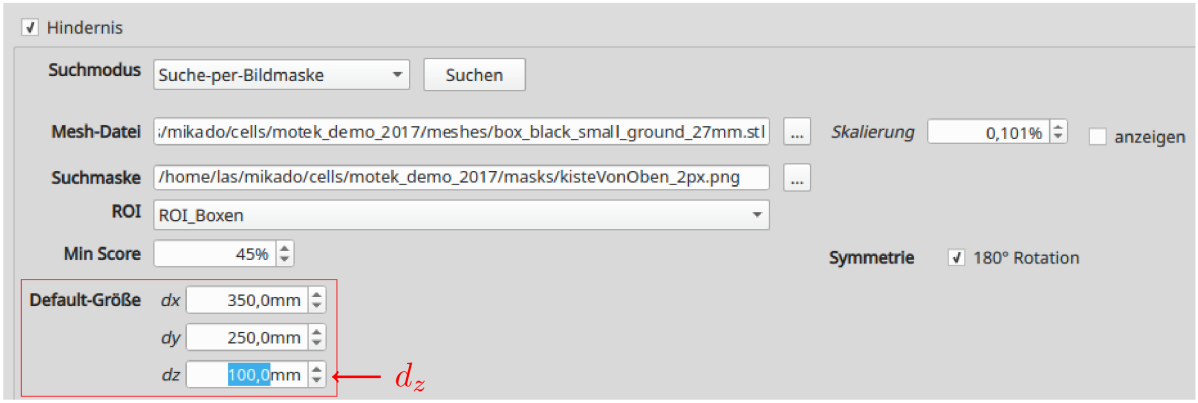

Abb. 43: Einstellung der Default-Größe einer Kiste

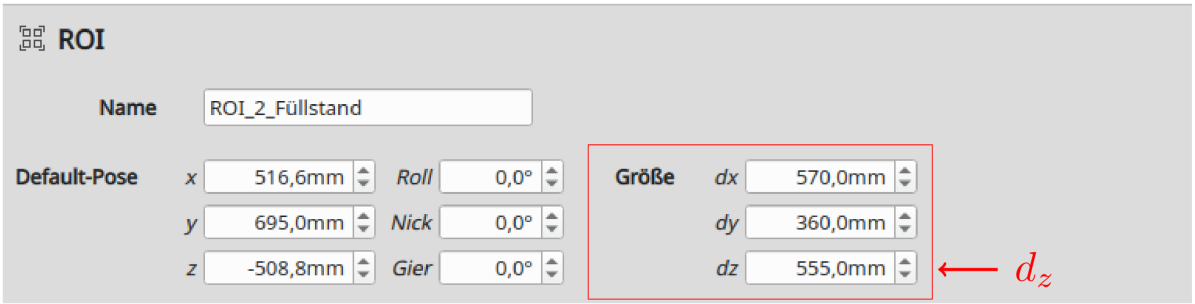

Abb. 44: Einstellung der Default-Größe einer ROI

2.2 Version 4

Neue Funktion `ROI` hinzugefügt, die bei der Suche nach Szenenobjekten benötigt wird. Bitte wählen Sie ein ROI-Objekt für jedes Szenenobjekt, das `Search-By-Image` verwendet.

2.3 Version 3

Neue Funktion `Symmetrien suchen` hinzugefügt, die bei der Suche nach Szenenobjekten verwendet wird. Standardmäßig deaktiviert.

2.4 Version 2

Die Standard-Dateiendung für Szenenobjekte hat sich geändert. Bitte speichern Sie Ihre Mikado-Szenenobjekt-Dateien ab sofort mit der Endung `*.scene`. Die Pfade für das Startprogramm in config/mikado.yaml müssen manuell bearbeitet werden.