

# Lokalisieren von 3D-Teilen für Bin-Picking

Studiengang: BSc in Elektrotechnik und Informationstechnologie  
Betreuer: Prof. Dr. Norman Urs Baier  
Experte: Dr. Enno de Lange (Johnson Electric)

Um Roboter in die Lage zu versetzen, Teile zu greifen, müssen diese Teile häufig von Hand vereinzelt und in Schablonen gelegt werden, welche dafür sorgen, dass die Teile immer an der gleichen Stelle liegen. „Pose Estimation“-Technologien ermöglichen es jedoch dem Roboter, die Teile selbstständig an beliebigen Orten zu greifen. Positionsschätzungs Algorithmen wie PPF und ICP wurden auf ihre Qualität und Einsatzbereiche getestet.

## Ausgangslage

Die aktuelle Situation der Robotik-Plattform ACROBA weist zurzeit einige Herausforderungen auf. Um Bauteile greifen zu können, müssen diese üblicherweise an vordefinierten Orten platziert werden, die hart kodiert sind. Um dieses Problem zu lösen, existieren bereits zwei Lösungsansätze. Der erste Ansatz verwendet einen RANSAC-Algorithmus und nutzt ein neuronales Netzwerk, um die Pose aus den 2D-Bildern zu schätzen. Der zweite Ansatz setzt auf das Projekt CNOS. Leider funktionieren beide Ansätze noch nicht zur vollsten Zufriedenheit. Daher wird derzeit auch die Verwendung von Halcon, eine weitere Softwarelösung, getestet, die jedoch sehr kostspielig ist. Ziel dieser Arbeit ist es, eine alternative Software zu entwickeln, die das gleiche Problem kostengünstiger lösen soll. Mit der Verwendung von Pose Estimation Algorithmen wie PPF und ICP soll deren Qualität und Einsatzbereich getestet werden.

## Ziele

Das Programm soll ein 3D-Matching zweier Punktwolken mithilfe von Positionsschätzungsalgorithmen durchführen. Eine der Punktwolken wird von einer 3D-Kamera erzeugt, während die andere aus einer STL-Datei des Objektes erstellt wird. Die Software soll eine möglichst exakte Position des Objekts bestimmen und diese Information an andere Programme weitergeben, damit das Objekt von der Robotik-Plattform ACROBA präzise aufgehoben werden kann. Darüber hinaus soll die Software in der Lage sein, Objekte

zu erkennen, die nicht perfekt ausgerichtet sind oder übereinander liegen. In solchen Fällen soll die Software automatisch die greifbare Position eines Objektes ermitteln und ausgeben. Ein weiterer wichtiger Aspekt ist die Dynamik der Software: Es soll möglich sein, das gesuchte Objekt zu ändern, ohne Eingriffe in den Code vornehmen zu müssen.

## Umsetzung

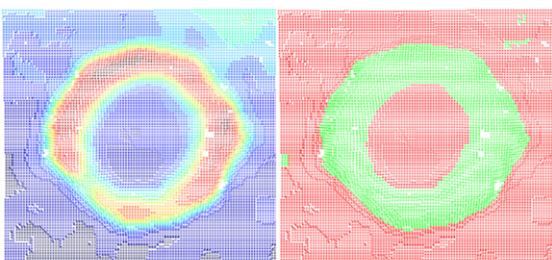
Zu Beginn wird eine STL-Datei des 3D-Objekts eingelesen mittels der Open3D-Bibliothek, wird dies in eine Punktwolke umgewandelt. Interne Punkte werden mit dem HPR herausgefiltert. Eine Punktwolke der Szene wird von der 3D-Kamera aufgenommen und mit der Objekt-Punktwolke mittels des Point Pair Feature (PPF)-Algorithmus verglichen. Zur Präzisionssteigerung folgt ein Iterative Closest Point (ICP)-Algorithmus. Die daraus resultierende Transformation Matrix gibt die Ausrichtung und Position relativ zur Kamera.

## Resultate und Ausblick

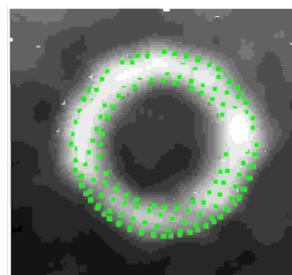
Das 3D-Matching-Programm hat Schwächen bei übereinanderliegenden Objekten und ist aufgrund der Python-Implementierung langsam. C++ könnte die Geschwindigkeit verbessern. Die verwendeten Intel 3D-Kameras haben eine geringere Präzision, was die Erkennung kleinerer Objekte erschwert. Eine leistungsfähigere 3D-Kamera sollte für mehr Genauigkeit und Zuverlässigkeit erwogen werden.



Serge Weidmann  
Embedded Systems  
serge@awinformatik.ch



Links: Punktwolke mit verbessertem Kontrast. Rechts: Gefilterte Punktwolke von Objekt und Hintergrund



3D-Matching des Objekts mit der Punktwolke