Introduction
Outline

- Motivation / Problem Statement
- Tool holder
- Sensor calibration
- Motion primitive
- Concatenation of clouds
- Segmentation
- Next possible pose
- Problems and Challenges
- Future Work
Motivation / Problem Statement

- Field of view is limited
  - Inside of cabinets, backsides of tabletop scenes
- Information exclusively from the front side
- More information with movable camera
  - 87% of scene with head mounted Kinect
  - 97% with the in-hand Sensor
Motivation / Problem Statement

- Field of view is limited by using only the inbuilt camera
  - Inside of cabinets, backsides of tabletop scenes
- Information exclusively from the front side
- More information with movable camera (Ch. Potthast 2011)
  - 87% of scene with head mounted Kinect
  - 97% with the in-hand Sensor
**Tool Holder**

- Design of the Tool
- 3D Printed
- Gripper center plate as mounting position
Calibration of Eye-in-Hand Camera

- Using camera_pose_toolkit package
- Based on the paper “Calibrating a multi-arm multi-sensor robot” by Pradeep, Konolige, Berger from 2010
- Both cameras observe a known object multiple times
- Modeled after the sparse bundle adjustment problem
Motion primitive
Registration of EIH-clouds

- Using ICP for registration
- Point to Point distance
- Two iterative steps
  - compute correspondences between the two scans
  - compute a transformation which minimizes distance between corresponding points
- Terminate when change in the error falls under a preset threshold
Registration of EIH-clouds

- **Linear ICP**
  - Reduces the squared error
  - Best results for our purpose (time and convergence)

- **Non linear ICP**
  - Based on Levenberg-Marquardt
  - Will converge even with bad initial position of cloud

- **Generalized ICP**
  - Euclidean distance plus uncertainty
  - Assumes that measured points have Gaussian failure model
  - MLE for iteratively estimated transformation
Next best pose

- Motion primitive works only for made-up scenes
- limit the set of possible camera poses according to sensor and robot model
- perform the raytracing from all poses from the above set of possible camera poses and determine occupied, free and unknown space
- compute the best pose by computing the entropy, e.g. as in “An Information Theoretic Approach for Next Best View Planning in 3-D Reconstruction” from Friedrich-Alexander University of Erlangen
Towards computation of next best pose

- Filter background
- Normal estimation
Towards computation of next best pose

- Filter background
- Normal estimation
- Segmentation of biggest plane
  - Perpendicular to z-axis of base link
  - Extract biggest cluster
  - Project to perfect plane
  - Reconstruct Convex hull
Towards computation of next best pose

- Filter background
- Normal estimation
- Segmentation of biggest plane
  - Perpendicular to z-axis of base link
  - Extract biggest cluster
  - Project to perfect plane
  - Reconstruct Convex hull
Towards computation of next best pose

- Filter background
- Normal estimation
- Segmentation of biggest plane
  - Perpendicular to z-axis of base link
  - Extract biggest cluster
  - Project to perfect plane
  - Reconstruct Convex hull
- Extract the object cluster using a polygonal prism
Next possible pose
Problems / Challenges

- Limited workspace of the robot’s arms
- Minimal recording distance of Kinect (600mm)

A: 400mm
B: 321mm
C: 50mm
Future Work

- Minimizing minimal recording distance
  - NYKO -> Kinect Zoom Lens
  - Near mode of Kinect

- Model real robot workspace to limit pose space

- Implementation of the raytracing

- Compute the best pose by computing the entropy
Where to get the code

- code.cs.tum.edu
- Seie2011
- Bismark
- Next best pose

Thank you for your attention, your help, and your support