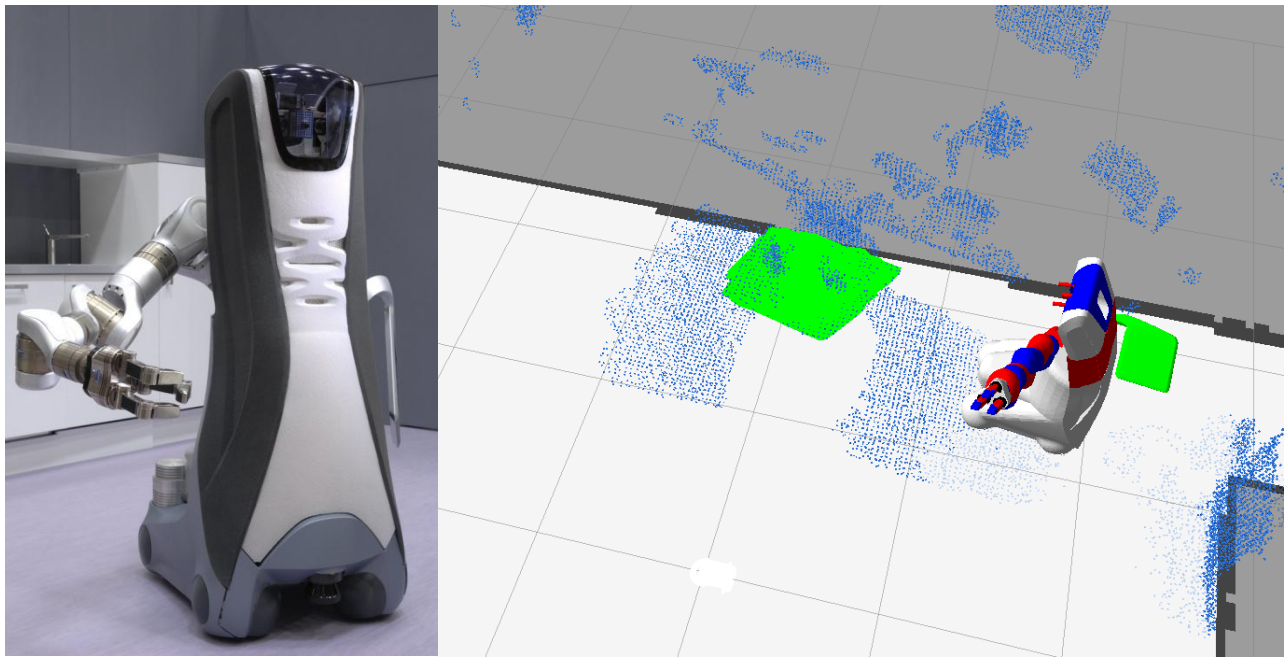
Field-of-view dependent registration of point clouds and incremental segmentation of table-tops using time-of-flight cameras

Dipl.-Ing. Georg Arbeiter

Fraunhofer Institute for Manufacturing Engineering and Automation IPA



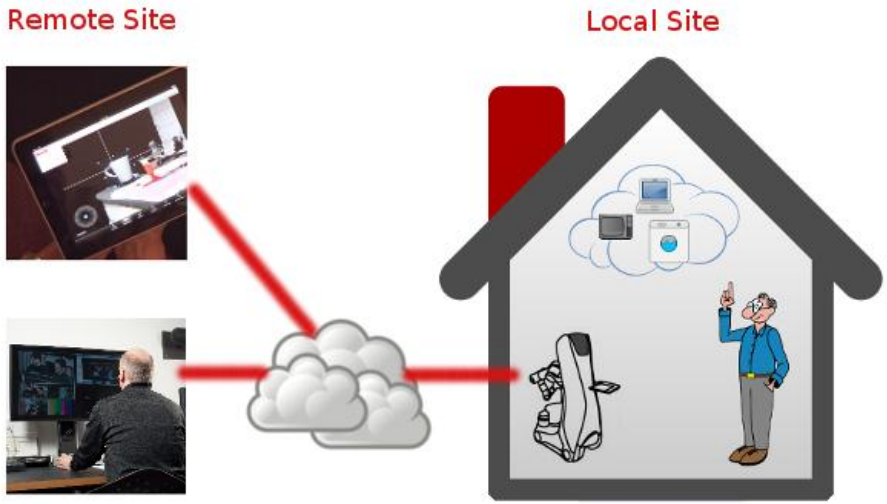
Fraunhofer IPA

Outline

- Application Scenario
- Requirements for Mapping
- System Architecture
- Mapping Process
- Experimental Results
- Outlook and Conclusion

Application scenario

- Tele-operated semi-autonomous robot in household environment
- Tasks like fetch-and-carry, monitoring
- Robot tries to solve household tasks autonomously
- Robot falls back to tele-operated mode for unexpected situations
- TOF cameras used



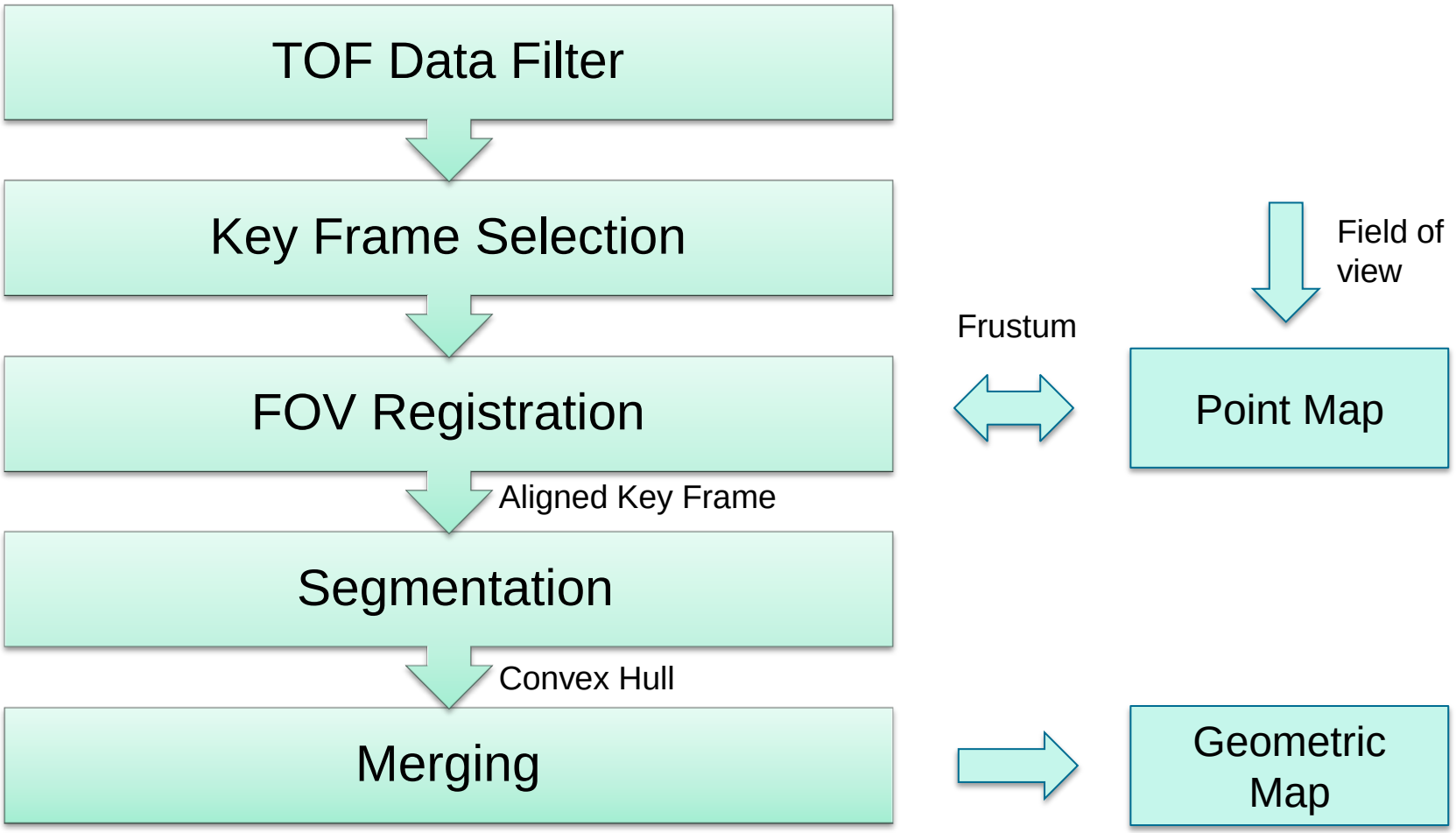
Fraunhofer IPA

Requirements for Mapping in Tele-Operation

- Applicability for
  - Navigation
  - Manipulation
  - Visualization

  both in autonomous and tele-operated mode
- Need for point and geometric map
- Map update must be
  - Fast
  - Incremental

System Architecture

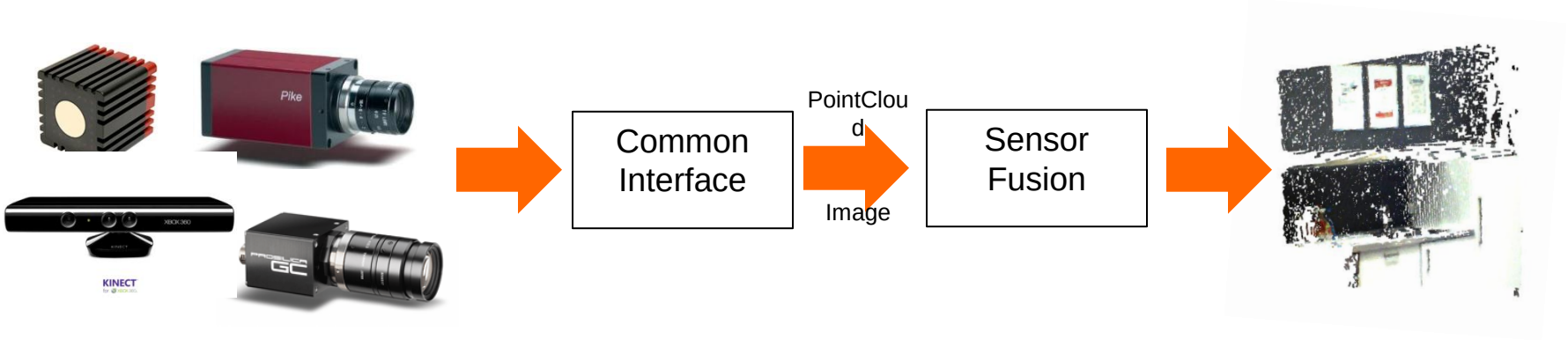


Outline

- Application Scenario
- Requirements for Mapping
- System Architecture
- Mapping Process
- Experimental Results
- Outlook and Conclusion

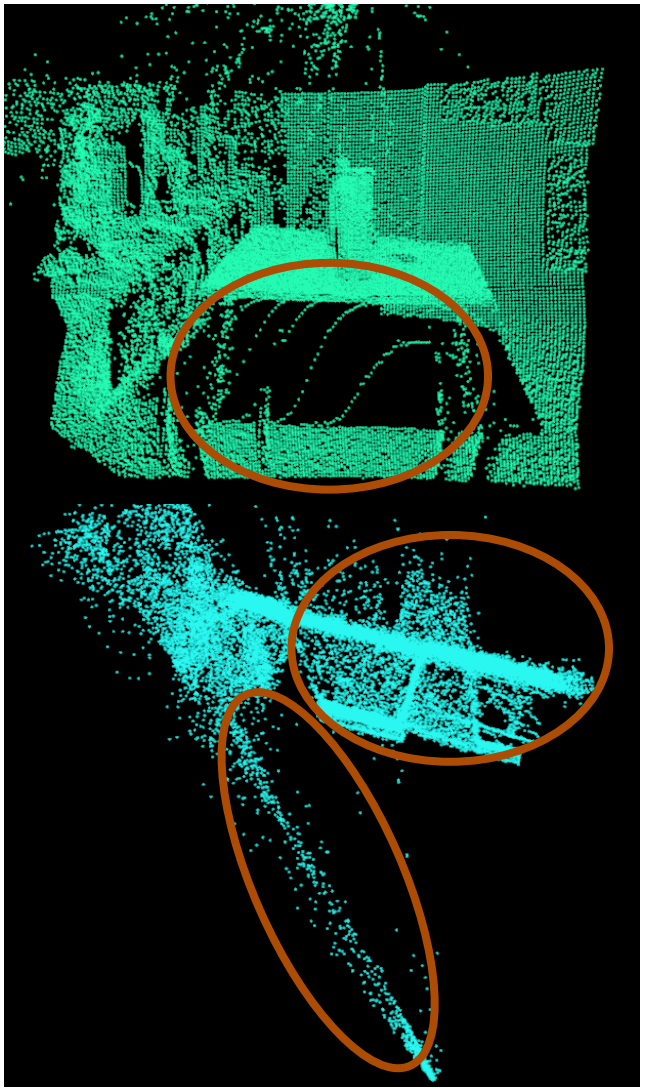
Sensor Fusion

- Combination of colour and depth values
- Transform depth values to color camera plain
- Output colored point cloud



Time-of-Flight Data Filter

- Time-of-flight errors
  - Jump edges
  - Gaussian noise
  - False depth values due to
    - Reflexions/ Transmission
    - Non-ambiguity range

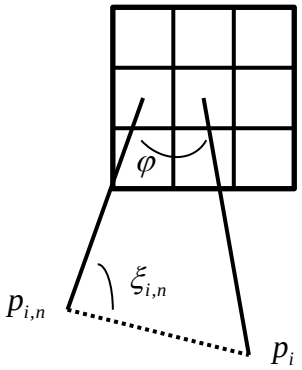


Filter Cascade

Jump Edge
- Neighbor relations
- Needs ordered data

Amplitude
- Thresholding

Stat. Outlier
- Remove remaining outliers



$$I_n < I_{th}$$

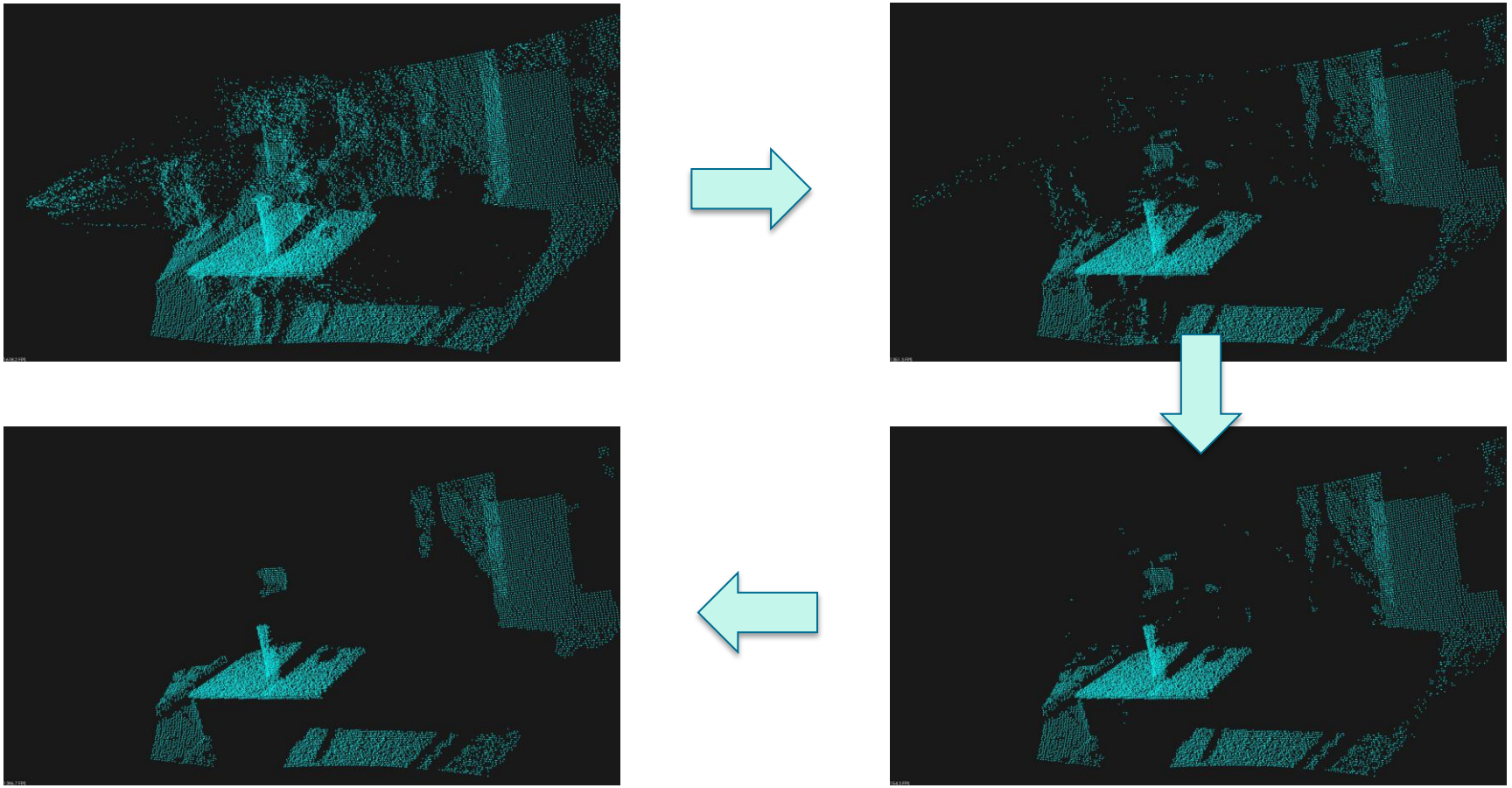
$$d(p_i) = \frac{1}{k}\sum_{n=1}^{k}\left\| p_i - p_{i,n} \right\|$$

$$d(p_i) < d_{th}$$

$$\xi_i = \max \arcsin\left(\frac{\left\| p_{i,n} \right\|}{\left\| p_{i,n} - p_i \right\|} \sin\varphi \right)$$

Fraunhofer IPA

Filter Cascade



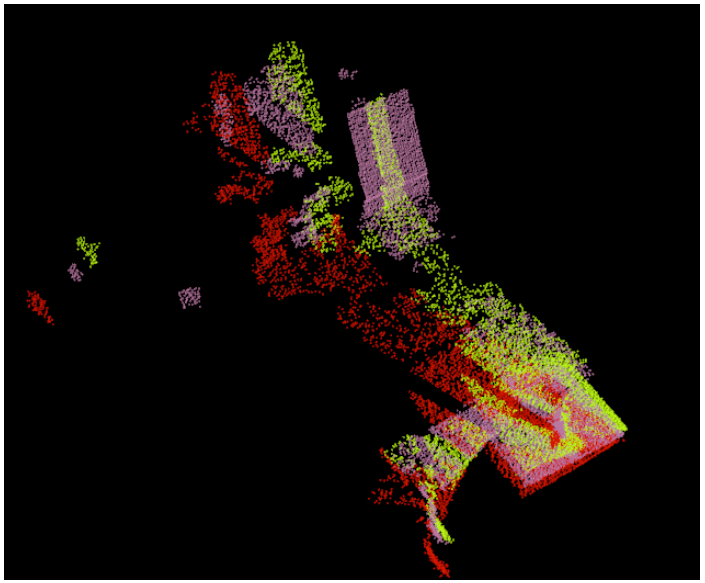
Fraunhofer IPA

Key frame selection

- 2-D localization using laser range finders
- Accept new frame only if robot moved significantly since last key frame
- Try to keep a certain overlap

24 frames

➔



3 key frames

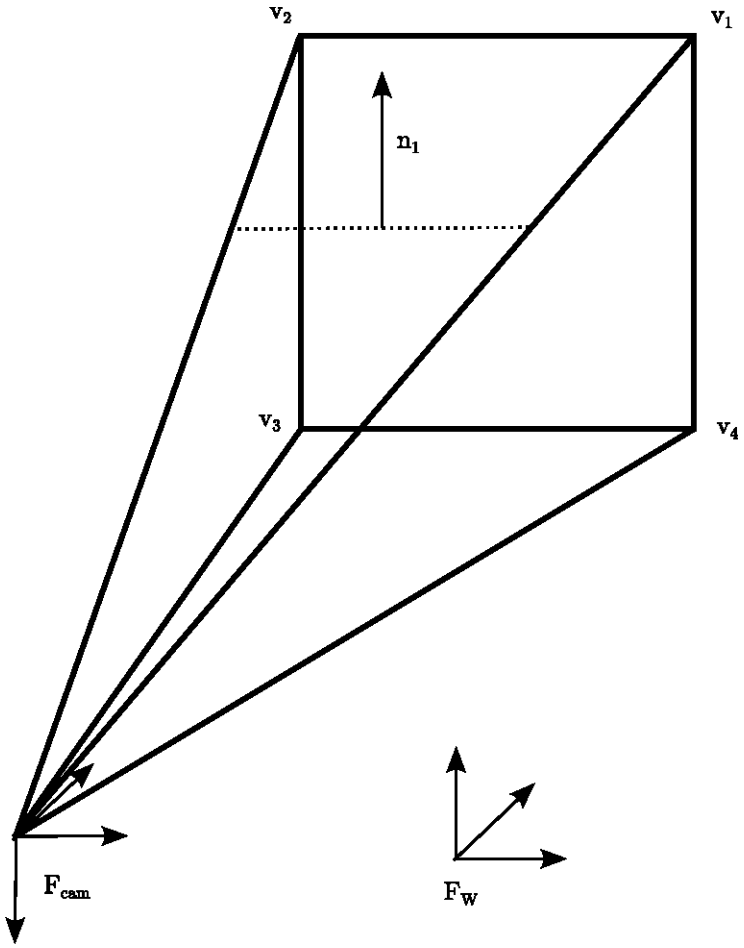
Registration using frustum ICP

- Only register key frames to the frustum part of the map
- Model frustum as pyramid
- Evaluate which points of map reside inside frustum
- Normal vector

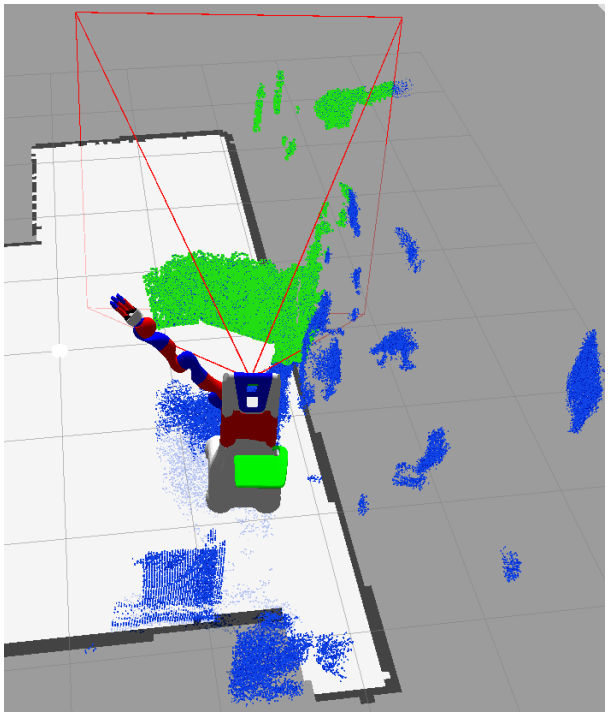
$$n_i = v_l \times v_m$$

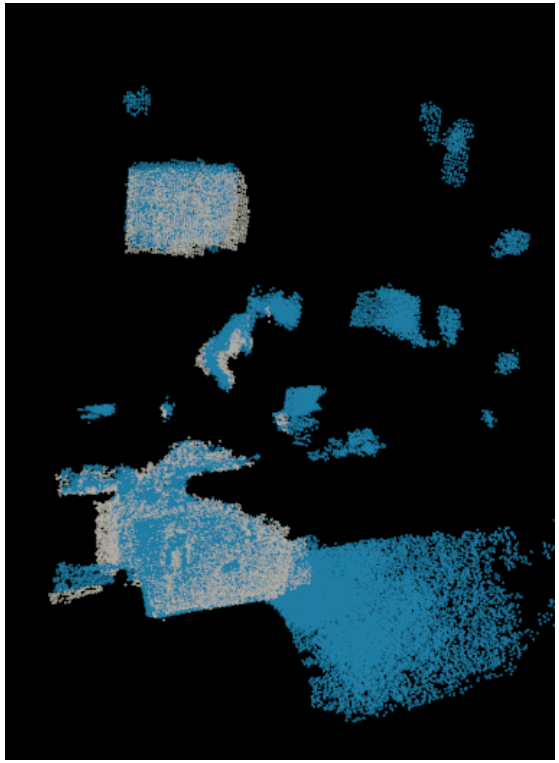
- Inlier criterion

$$(x_p - x_0)n_{i,x} + (y_p - y_0)n_{i,y} + (z_p - z_0)n_{i,z} < 0$$

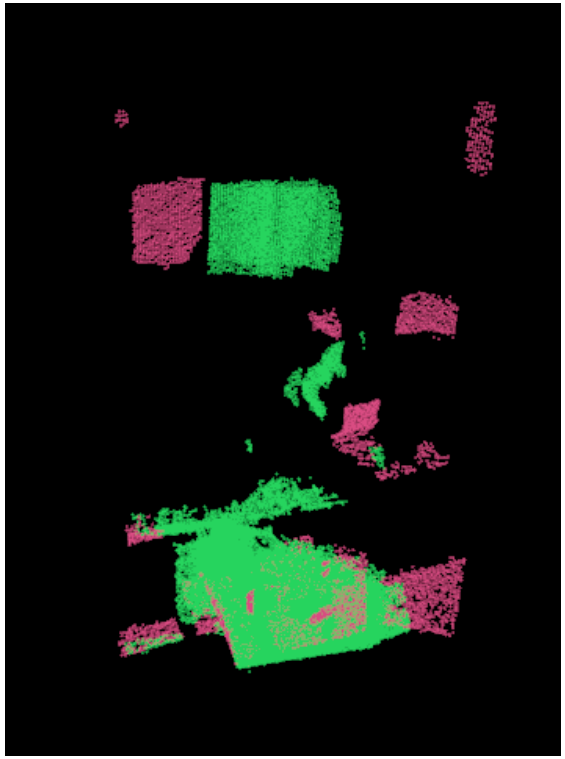


Registration using frustum ICP





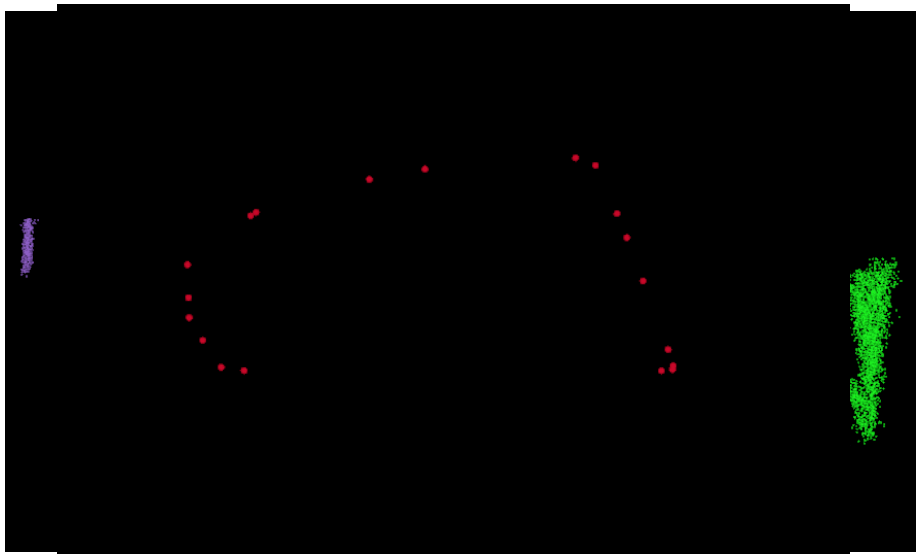
Map (blue) and frustum (grey)



Frustum (green) and new point cloud (pink)

Planar Segmentation

Eucledian Clustering → Normal Estimation → Plane Fitting → Inlier Projection → Convex Hull



Hull Merging

- Separating axes theorem
- Linear equation of perpedicular line

$$\vec{x} = \vec{p}_n + \lambda \begin{pmatrix} -\left(p_{n+1,y} - p_{n,y}\right) \\ p_{n+1,x} - p_{n,x} \end{pmatrix}$$

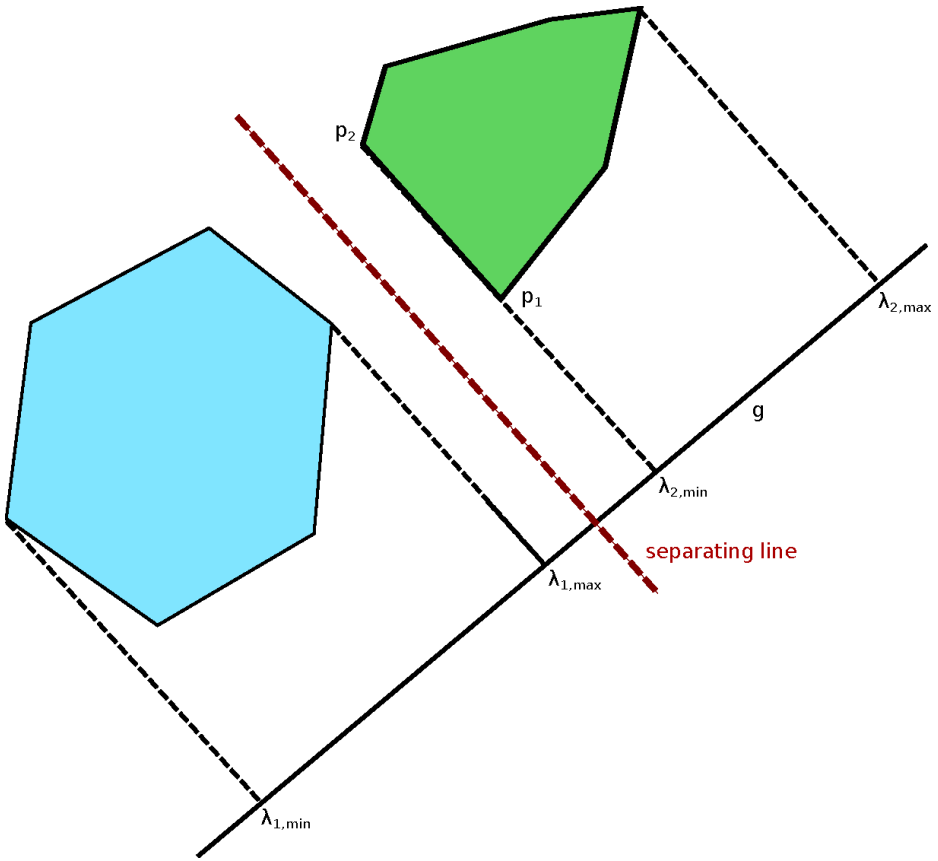
- Project points on line

$$(\vec{s}_i - \vec{q}_i) \circ \vec{g} = 0$$

- Decide if line is separating

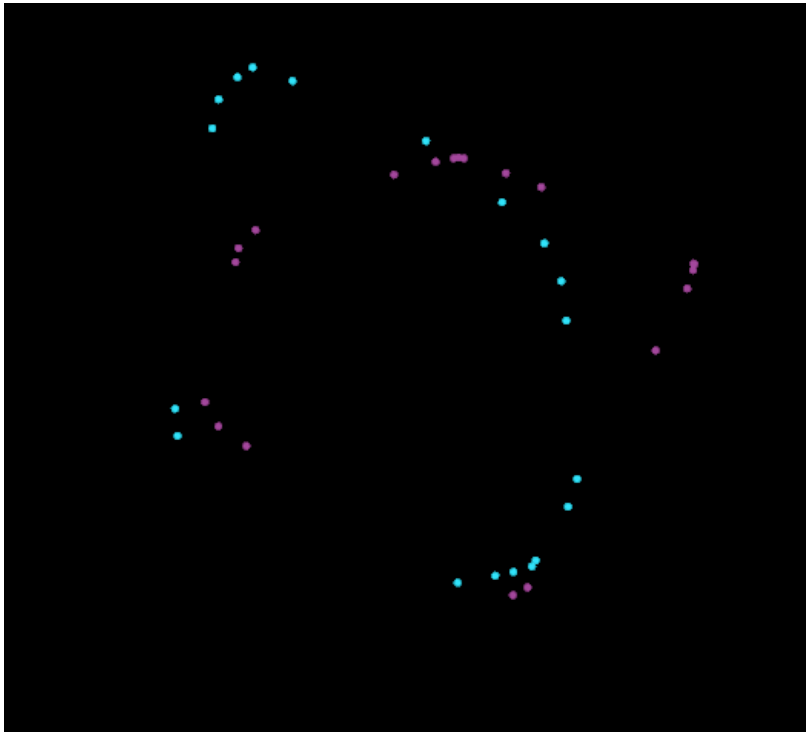
$$[\lambda_{1,\min}, \lambda_{1,\max}] \cap [\lambda_{2,\min}, \lambda_{2,\max}] = \varnothing$$

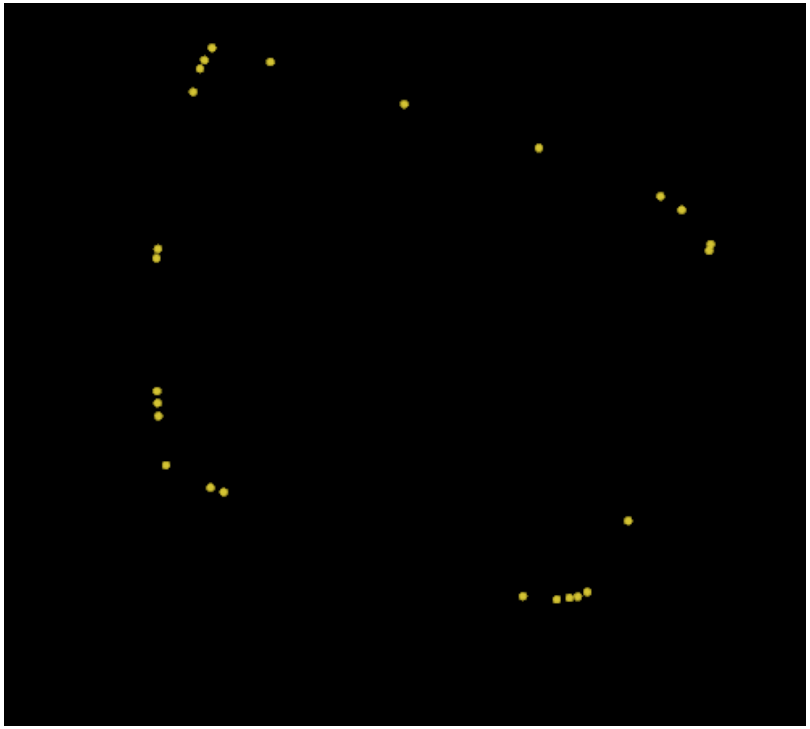
- Merge hulls if intersecting
  - Concat hulls
  - Calculate resulting hull



Fraunhofer IPA

Hull Merging





Outline

- Application Scenario
- Requirements for Mapping
- System Architecture
- Mapping Process
- Experimental Results
- Outlook and Conclusion

Care-O-bot 3

Light-Weight-Arm
(7 DOF)

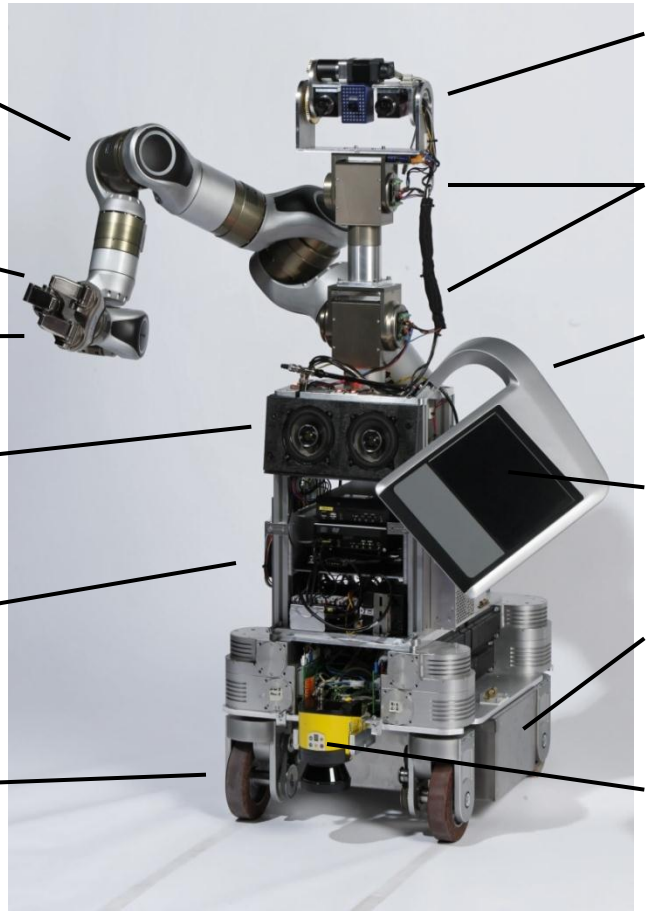
Gripper
(7 DOF)

Tactile sensors

Speakers

Computer Bay
(3-5 PCs)

Omnidirectional platform
(4x 2 DOF)



Sensor head
(1 DOF, stereo and 3D-ToF)

2 PT-Units
(2x 2 DOF)

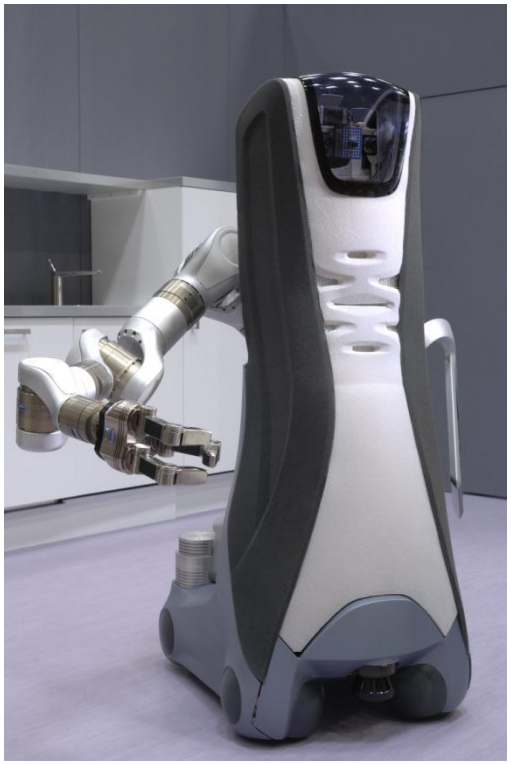
Tray
(1 DOF)

Touch screen

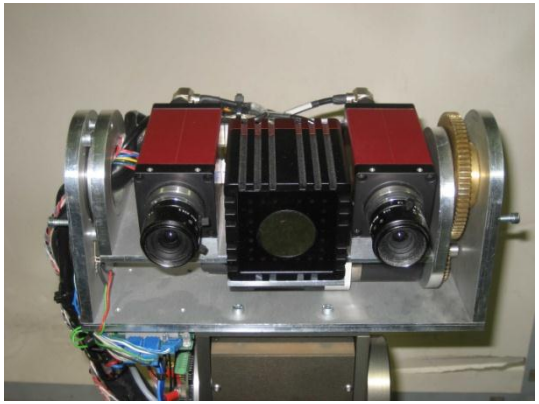
Battery

3 Laser scanners
(front, back, top of mobile base)

Experimental Results



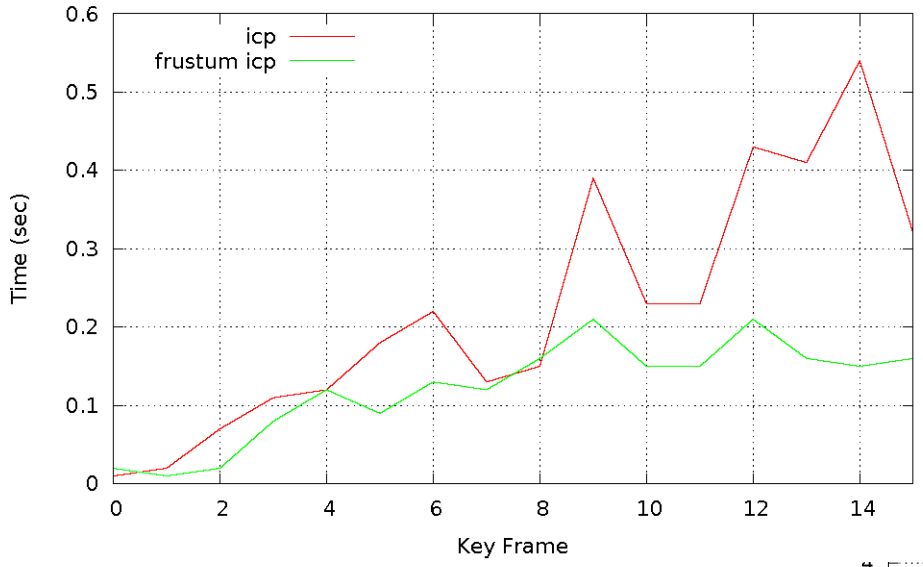
Care-O-bot



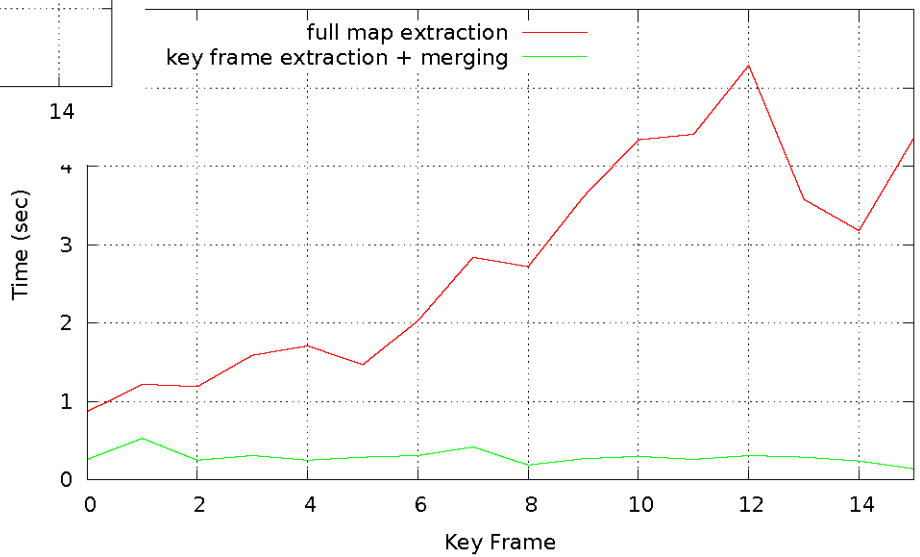
Sensor head

Test site

Processing time registration and segmentation

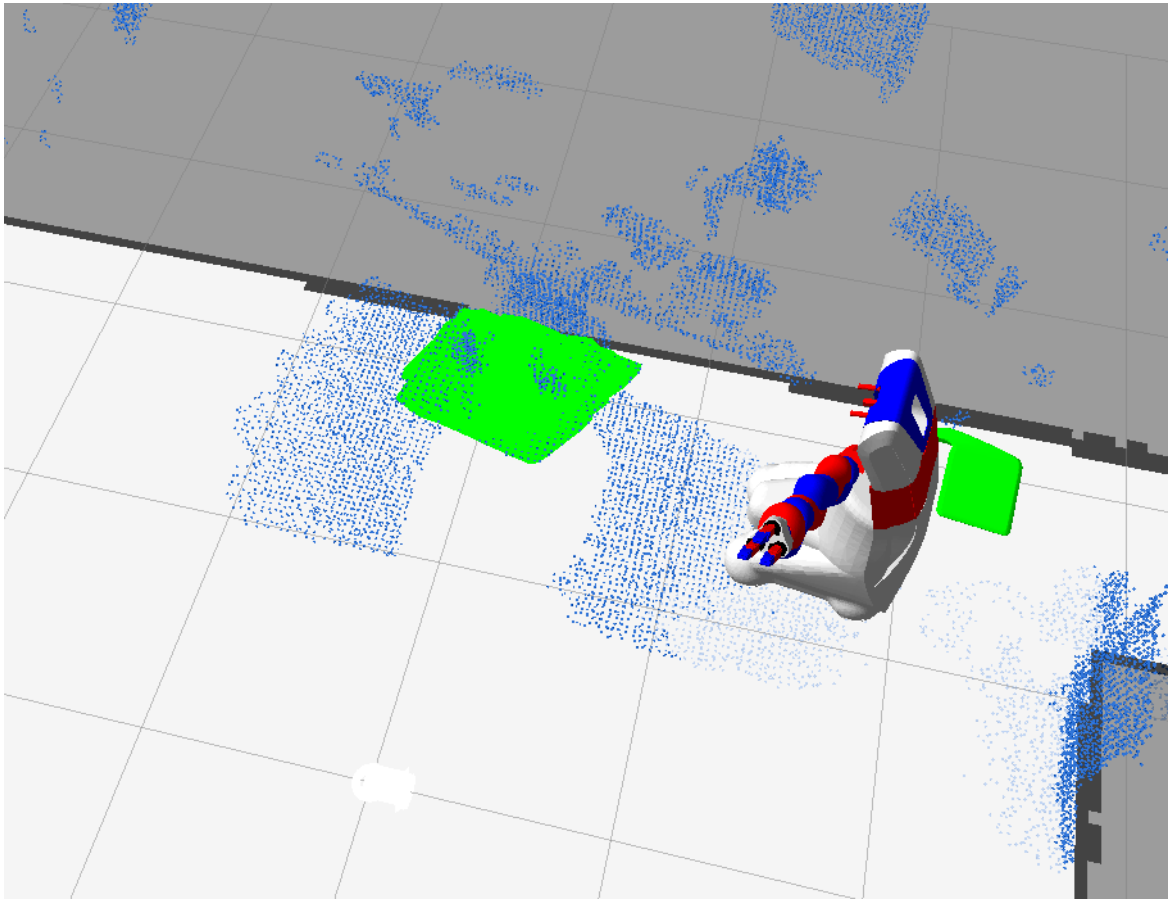


- Robot driven around table manually
- 2-D localization used
- 15 key frames



Fraunhofer IPA

Experimental Results



Application: Assisted Object Detection

- Table-top as region of interest
  - Augment in user interface
- Further investigation of ROI
- Segmentation of point cloud above table
- Clustering
- Input for object detection

Conclusion and Outlook

- Filter cascade for TOF data
- FOV dependent registration
- Incremental segmentation and merging

Future work

- Extraction of additional geometric objects
  - Perpendicular planes
  - Edges
- Point features
- Appropriate visualization of geometric map in user interface