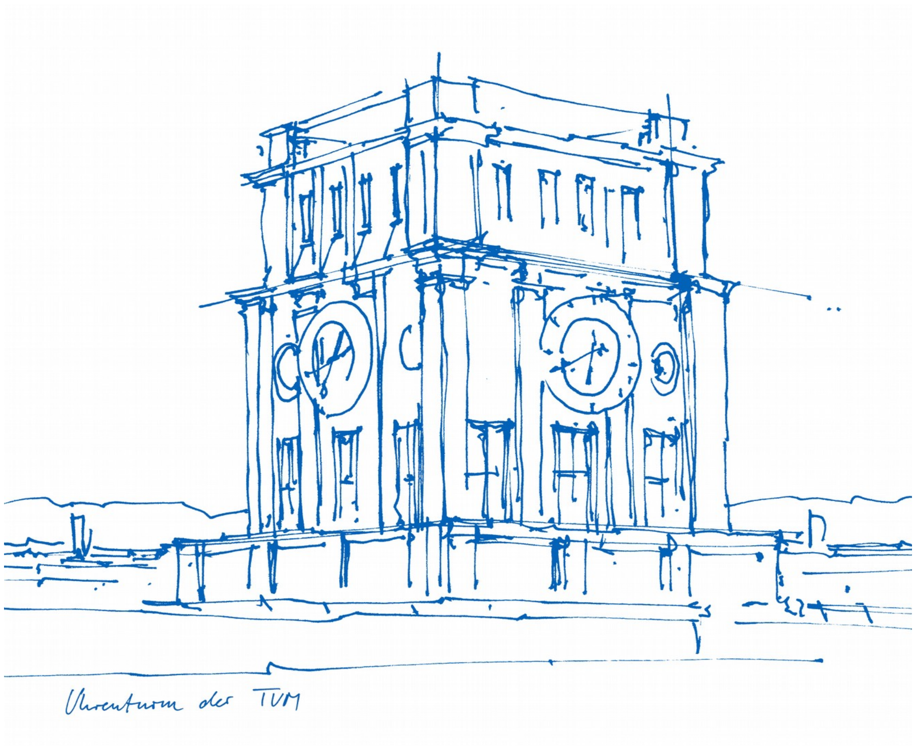
Bachelor’s Thesis: 3D Scene Reconstruction for 2D Object Recognition

Kanstantsin Tkachuk

Technische Universität München

Fakultät für Informatik

Garching bei München, 26. August 2019



From 3D to 2D: why and how?

Tasks in robotics are in general harder in 3D than 2D:

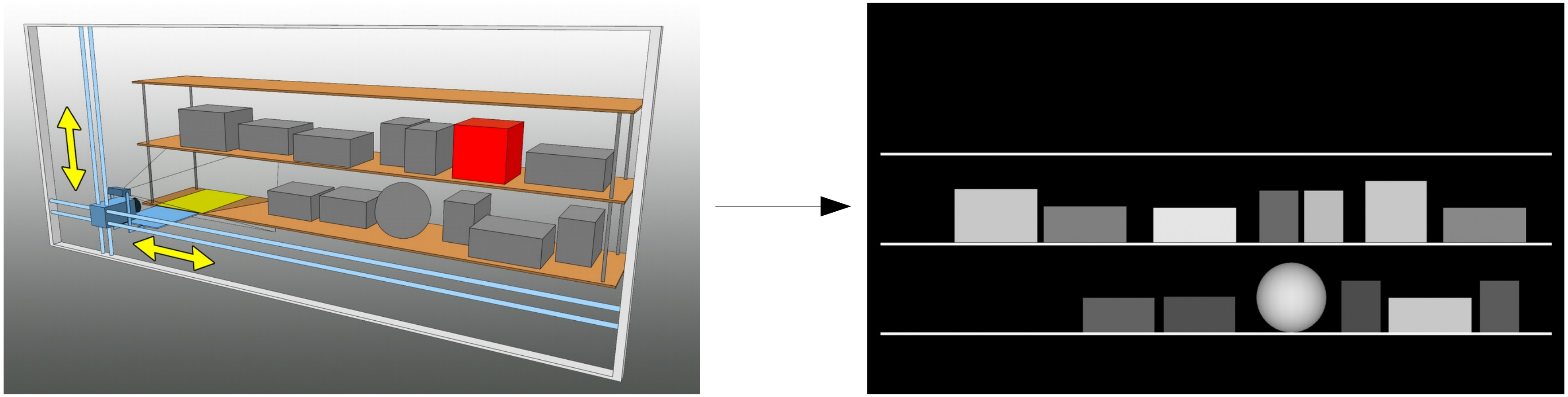
- localization
- object recognition
- path planning

Reason: a body in 3D-space has 6 degrees of freedom vs. 3 degrees of freedom in 2D.

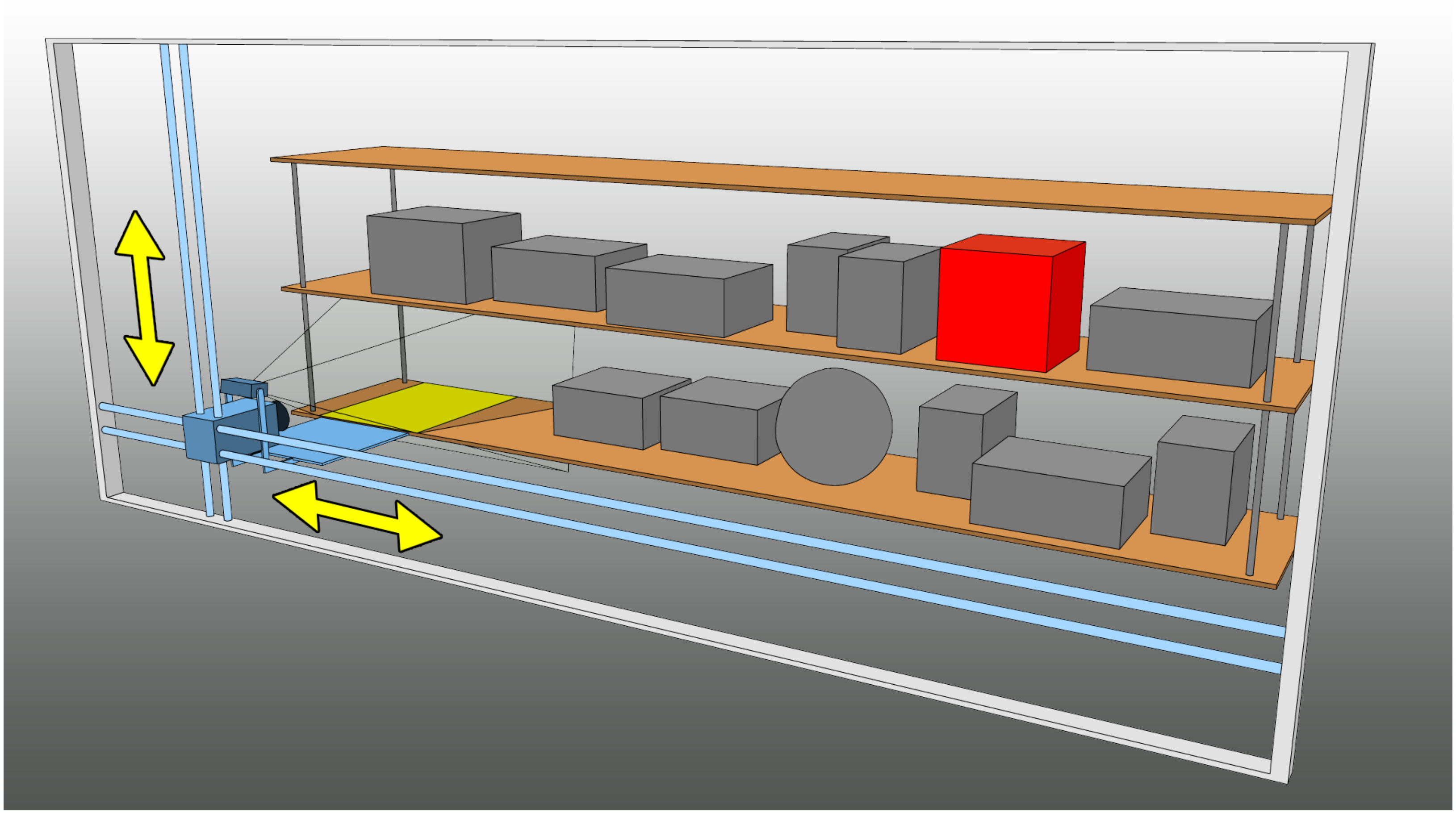
From 3D to 2D: why and how?

Useful fact: some problems in 3D can be reduced to an equivalent problem in 2D.

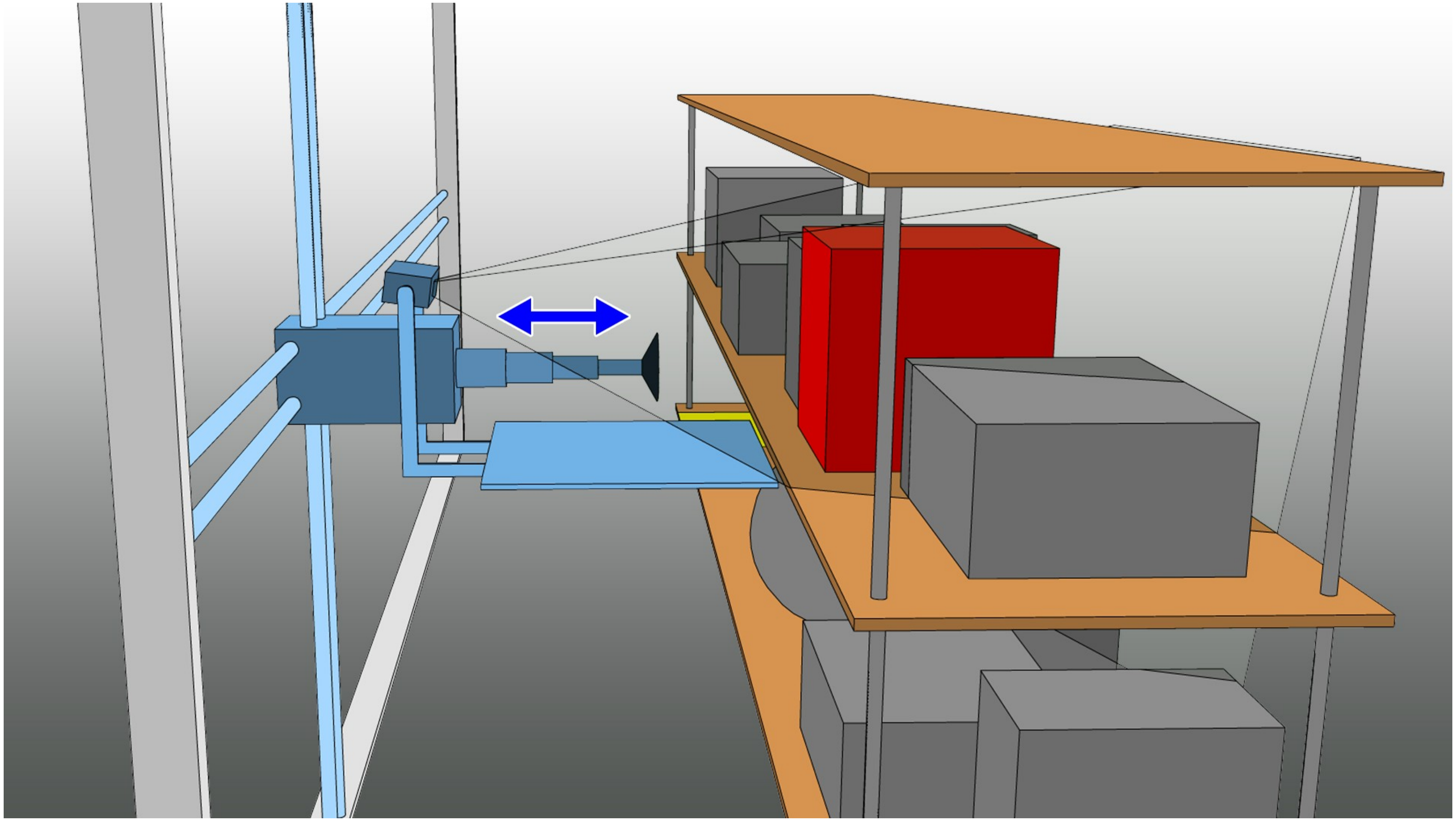
Specifically: problems for objects that have 3 DOF in the 3D-space instead of 6.



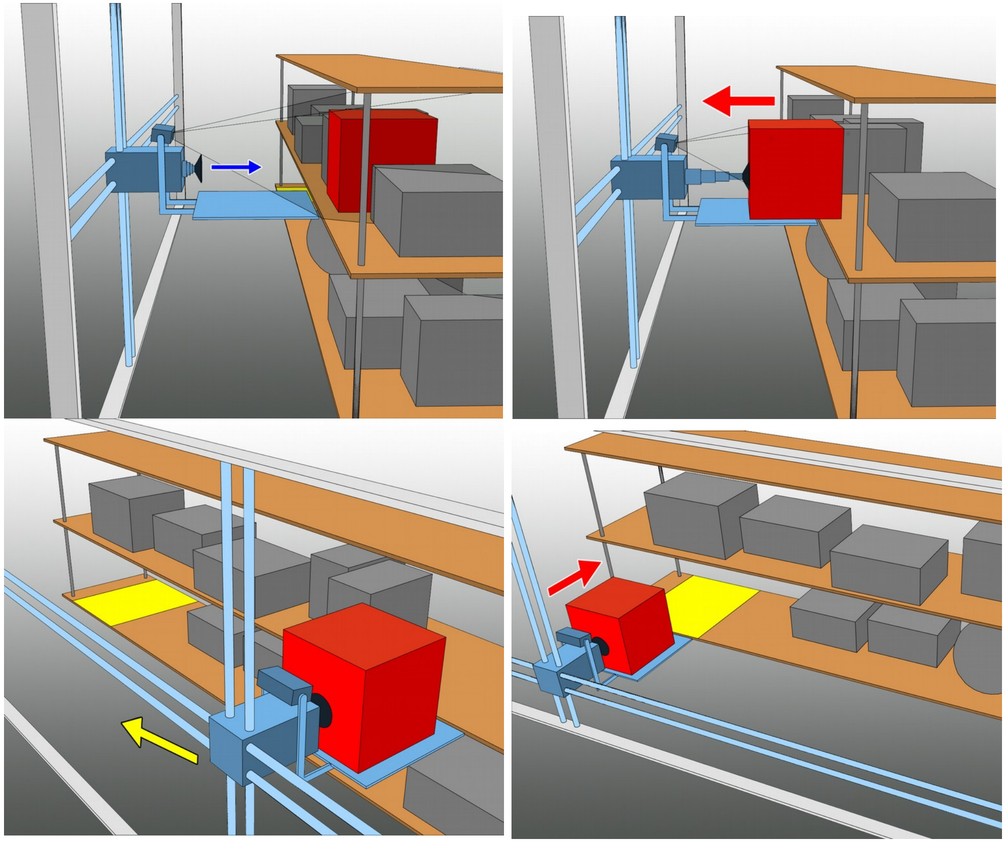
Problem setting: generalized planar robotic manipulator



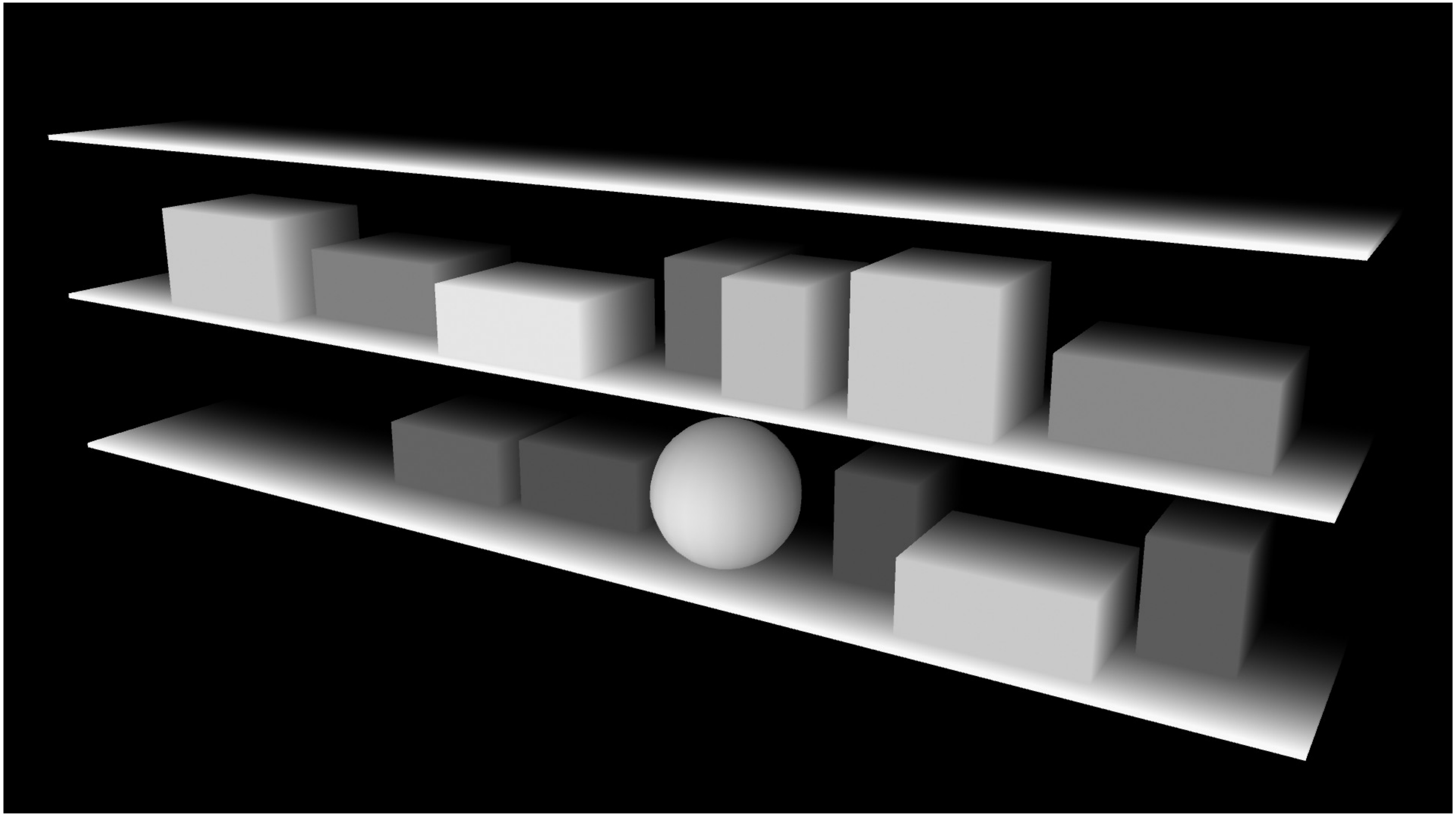
Problem setting: generalized planar robotic manipulator



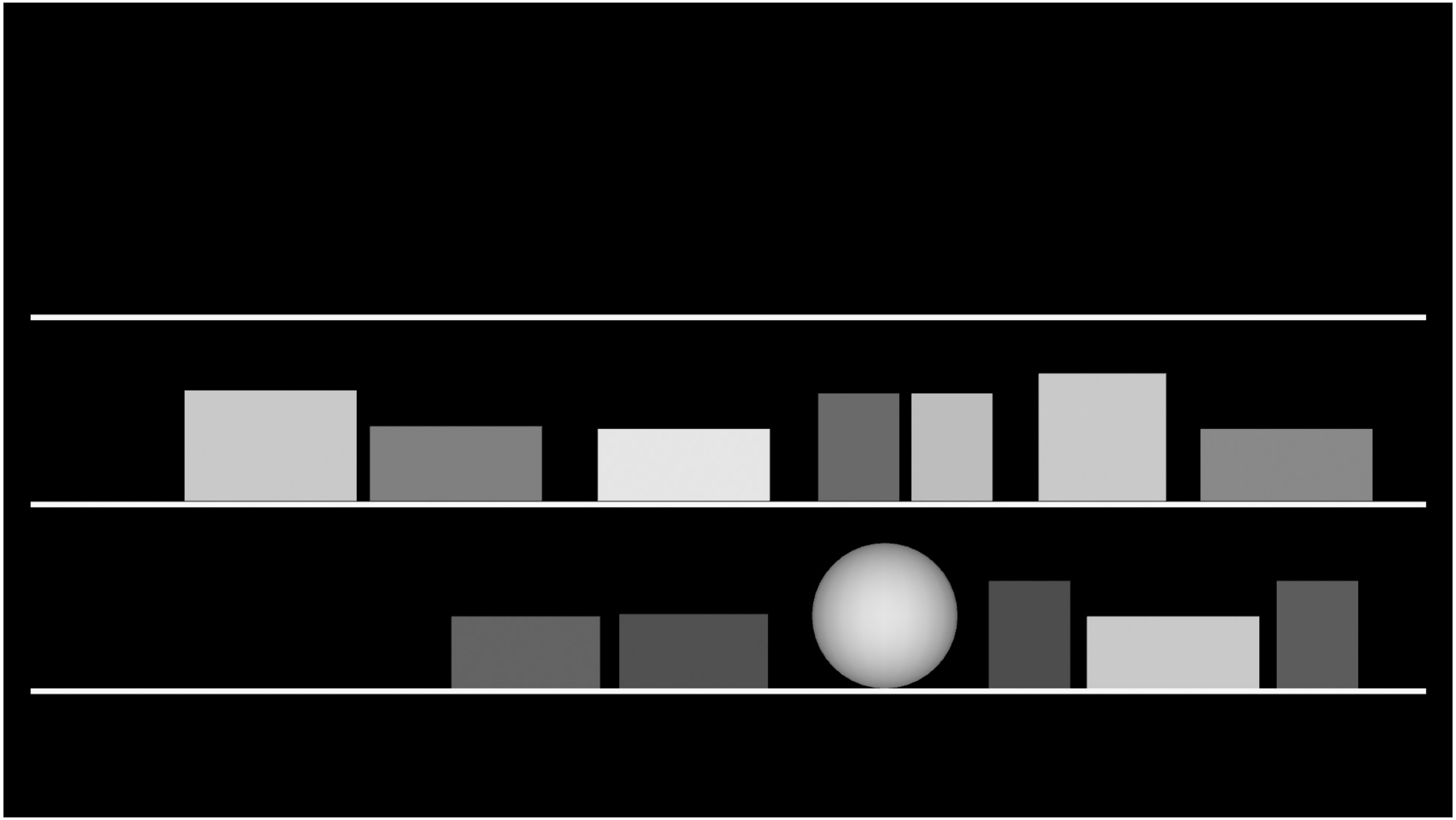
Problem setting: generalized planar robotic manipulator



Relevant information: frontal distances to objects



Relevant information in 2D-representation

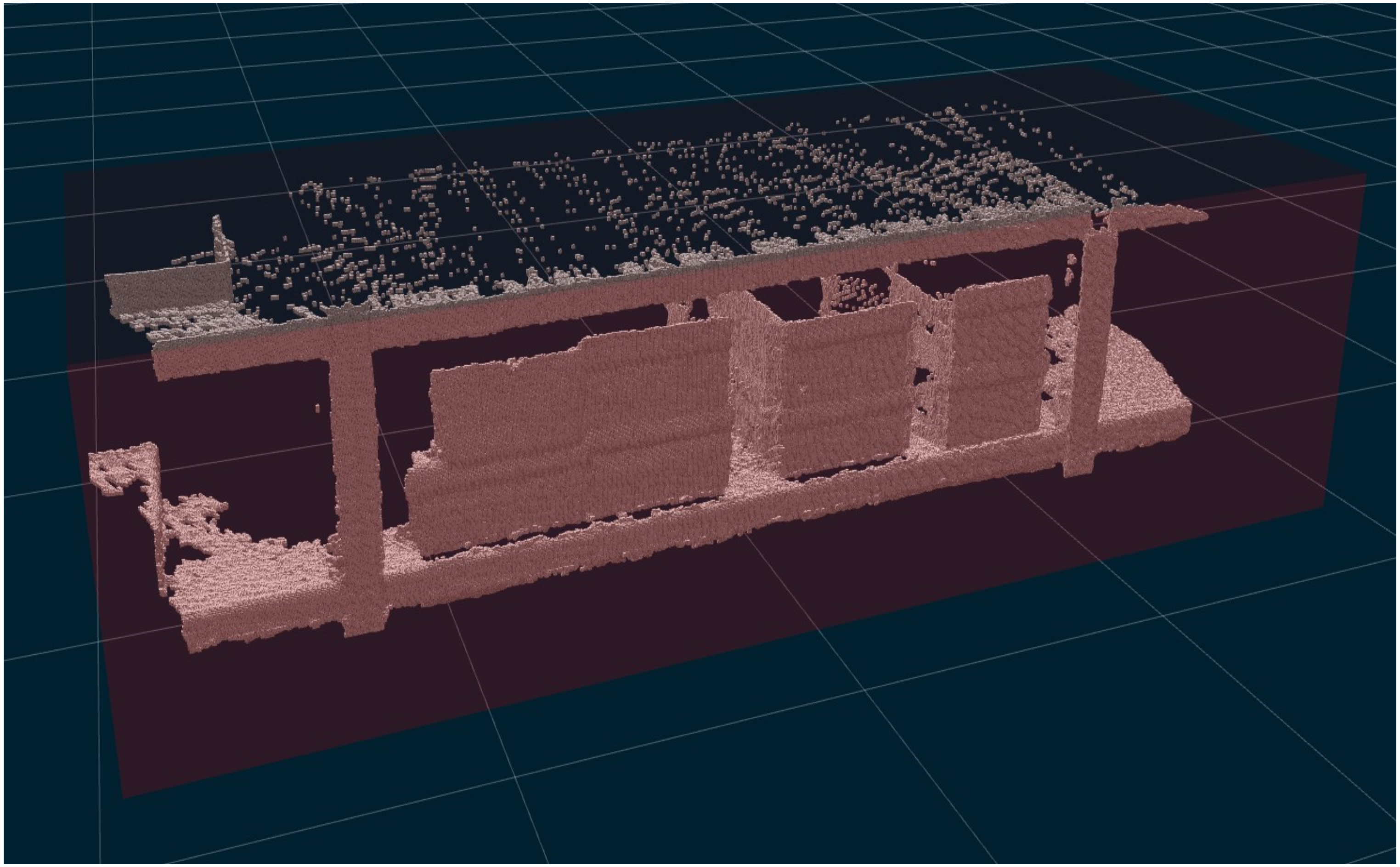


Real-life robot: TORU 5 by Magazino GmbH

TORU 5: manipulator

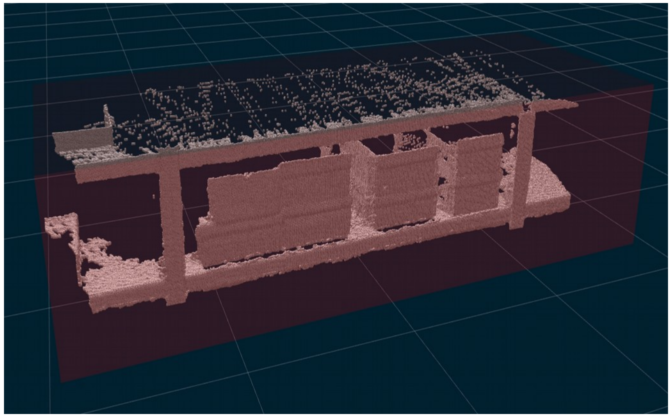
TORU 5: relevant information for object detection

TORU 5: 3D surface representation



From 3D to 2D: why and how?

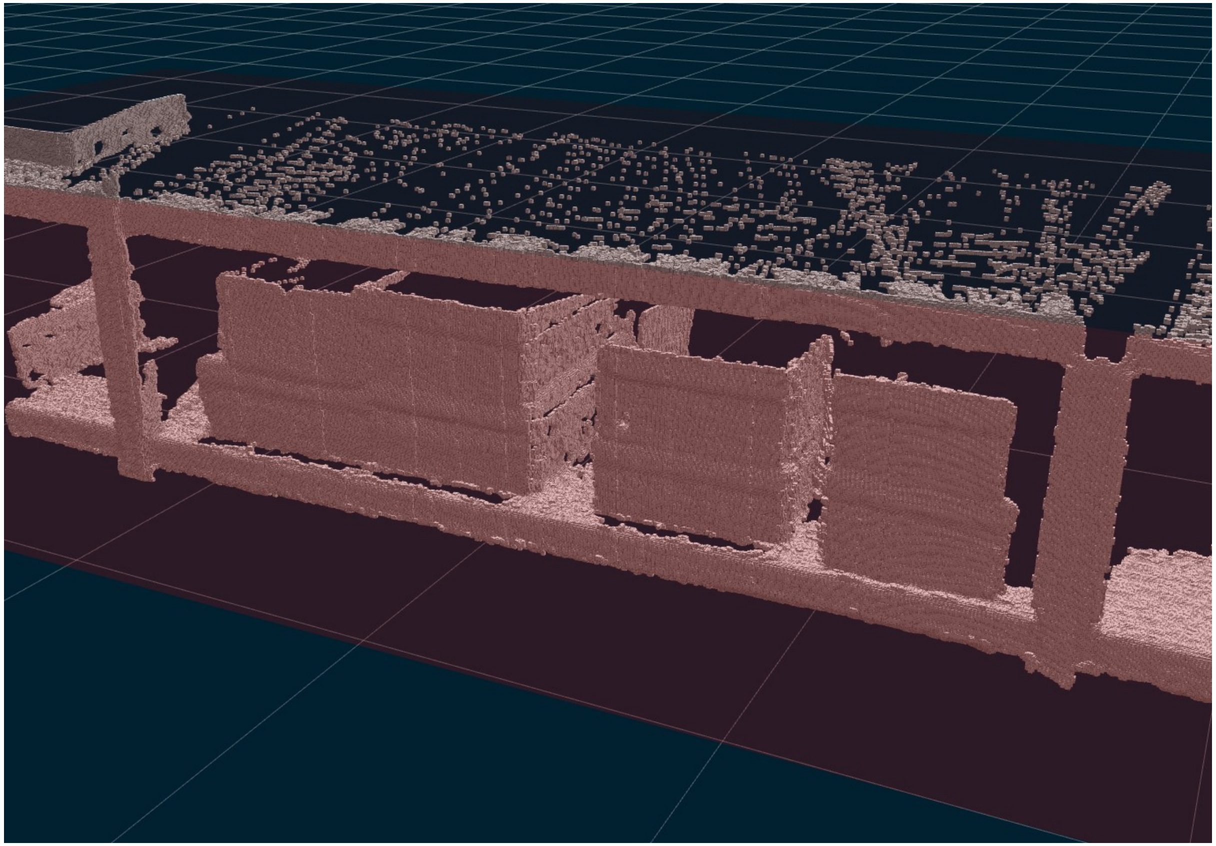
Goal: reconstruct the surface in 3D in order to create an 2D representation.



Advantage: the reduced problem can be solved much more efficiently.

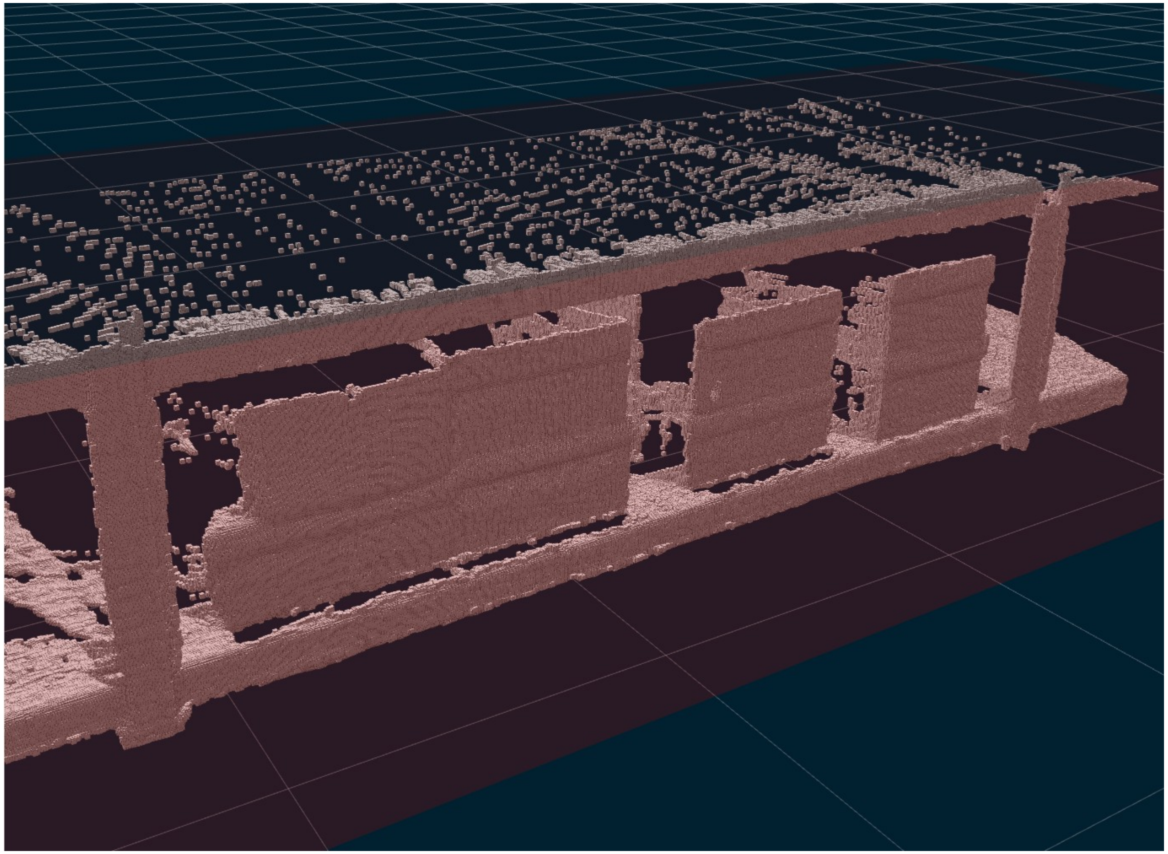
Disadvantage: limited sensor view

The sensor cannot “see” the whole shape due to reflections



Disadvantage: limited sensor view

The sensor cannot “see” the whole shape due to reflections



Goal: find a suitable reconstruction algorithm

A suitable for the given setting 3D surface reconstruction algorithm must:

- produce good 2D representations
- be robust against the incompleteness of the sensor data
- be computationally efficient

These requirements are the criteria against which the performance of the examined algorithms should be evaluated.

Assisted approach vs. KinectFusion approach

Assisted approach uses external sensor pose estimations to update the reconstructed surface with new depth data:
+ easy computations
- strongly affected by errors in external localization data

KinectFusion approach uses the KinectFusion algorithm to calculate the movement of the sensor between two frames:
+ self-sufficient: only needs the input from the sensor
- might be affected by incomplete sensor data: point cloud matching errors

The algorithms are evaluated in regard to the quality of the produced 2D representations.

KinectFusion approach

KinectFusion algorithm is an algorithm for 3D surface reconstruction using depth cameras.

“KinectFusion enables a user holding and moving a standard Kinect camera to rapidly create detailed 3D reconstructions of an indoor scene. Only the depth data from Kinect is used to track the 3D pose of the sensor and reconstruct, geometrically precise, 3D models of the physical scene in real-time.”

Source: original paper “Kinectfusion: real-time 3D reconstruction and interaction using a moving depth camera” by S. Izadi , D. Kim , O. Hilliges , D. Molyneaux , R. Newcombe , P. Kohli , J. Shotton , S. Hodges , D. Freeman , A. Davison , A. Fitzgibbon.

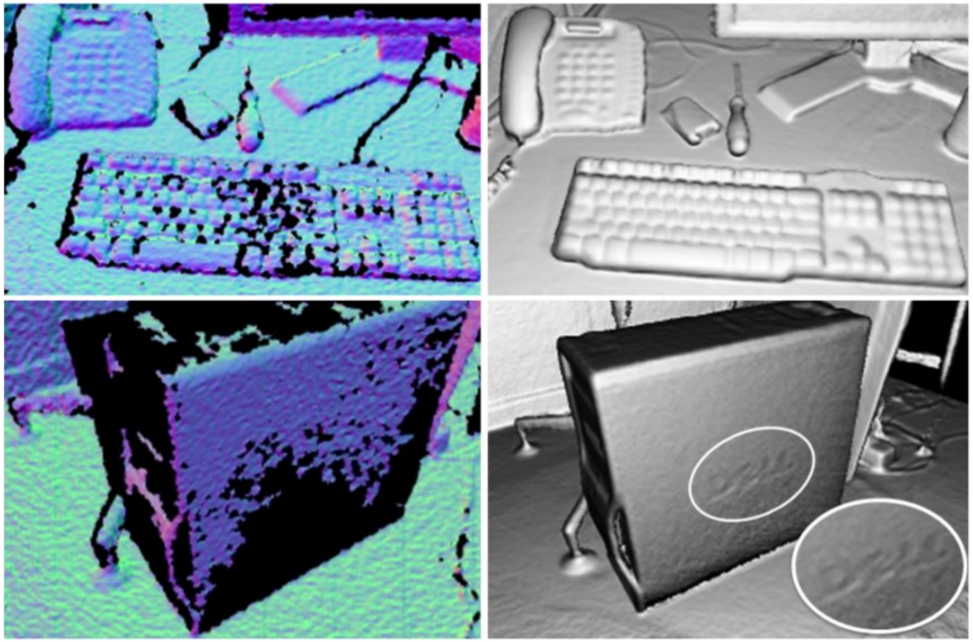


Figure 3: Left: Raw Kinect data (shown as surface normals). Right: Reconstruction shows hole filling and high-quality details such as keys on keyboard, phone number pad, wires, and even a DELL logo on the side of a PC (an engraving less than 1mm deep).

KinectFusion approach

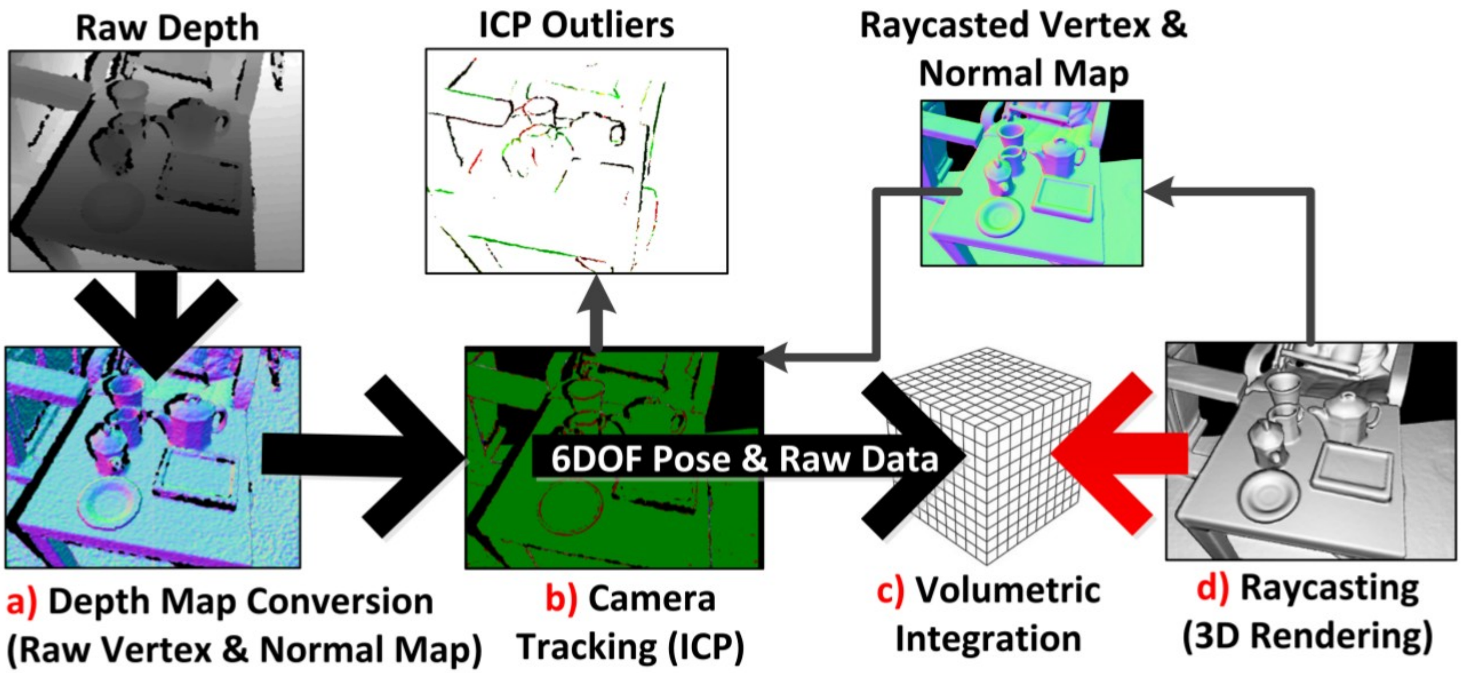
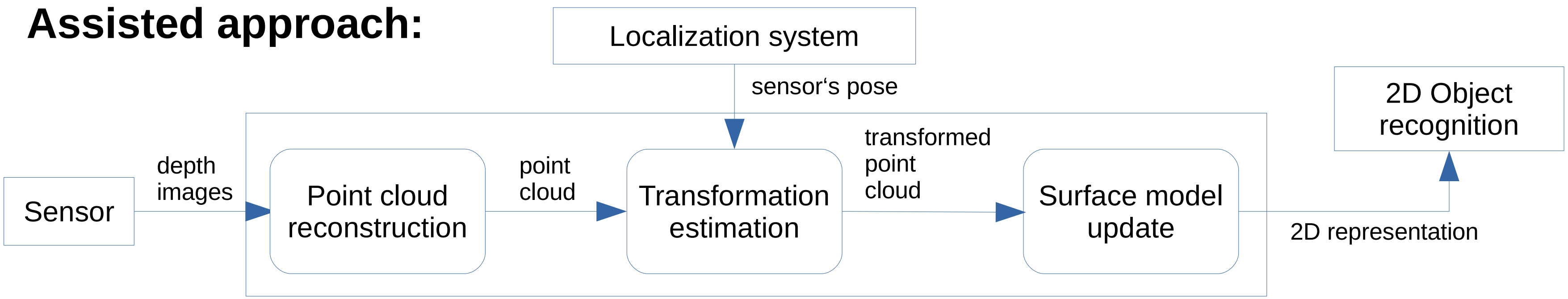


Figure 11: Overview of tracking and reconstruction pipeline from raw depth map to rendered view of 3D scene.

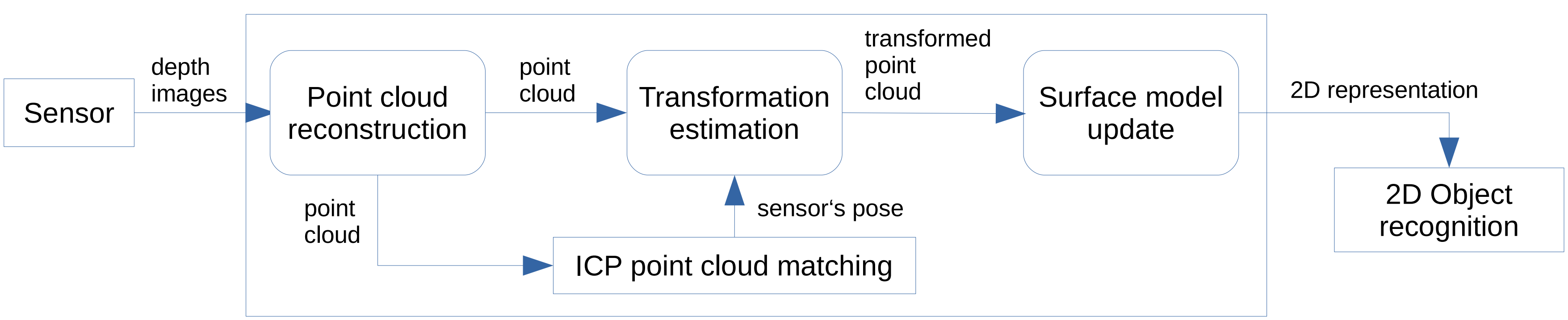
Source: original paper "Kinectfusion: real-time 3D reconstruction and interaction using a moving depth camera" by S. Izadi , D. Kim , O. Hilliges , D. Molyneaux , R. Newcombe , P. Kohli , J. Shotton , S. Hodges , D. Freeman , A. Davison , A. Fitzgibbon.

Reconstruction pipeline

Assisted approach:

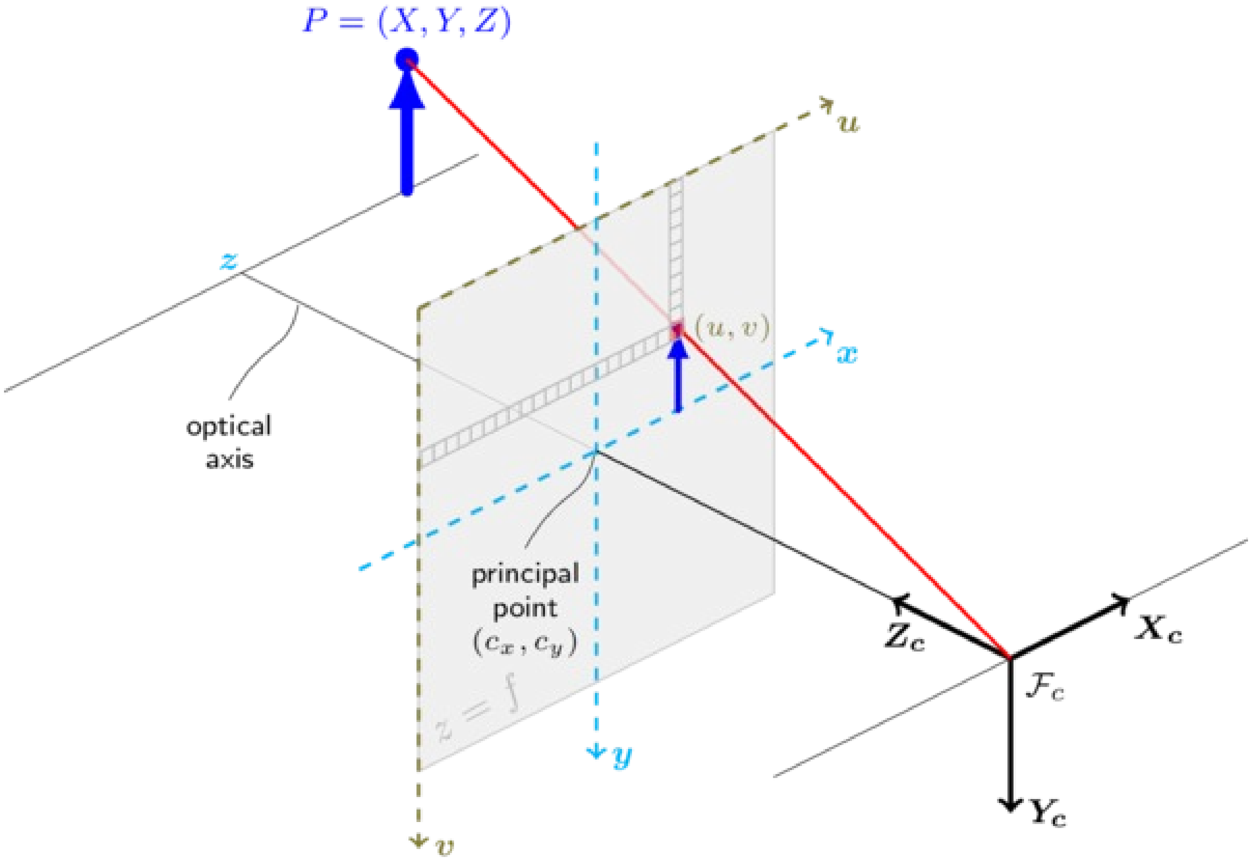


KinectFusion approach:



Point cloud reconstruction

Point cloud reconstruction: pinhole camera model



Source: OpenCV documentation, https://docs.opencv.org

Point cloud reconstruction: pinhole camera model

Intrinsic calibration matrix **K:**

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

+

Depth image **d:**

$$d(x, y)$$

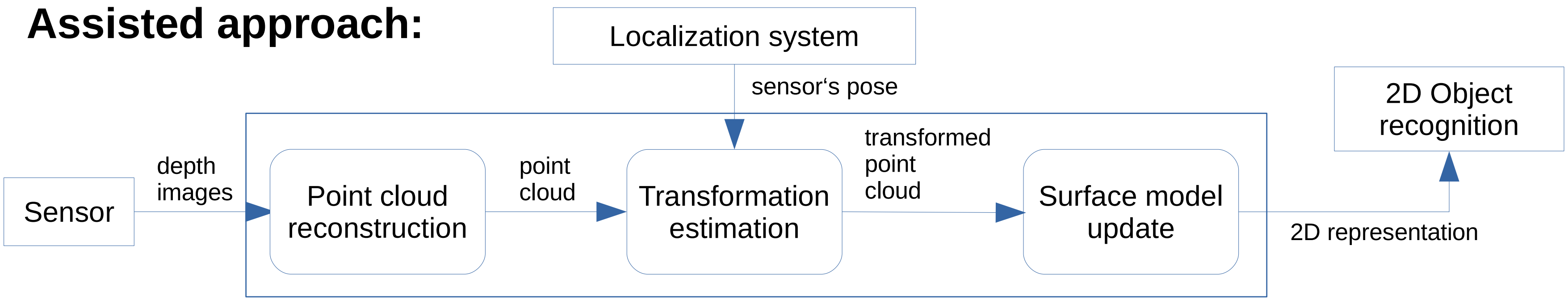
→

Point cloud **P**:

$$P_{x,y} = \begin{bmatrix} \frac{d(x,y)}{f_x}(x - c_x) \\ \frac{d(x,y)}{f_y}(y - c_y) \\ d(x,y) \end{bmatrix}$$

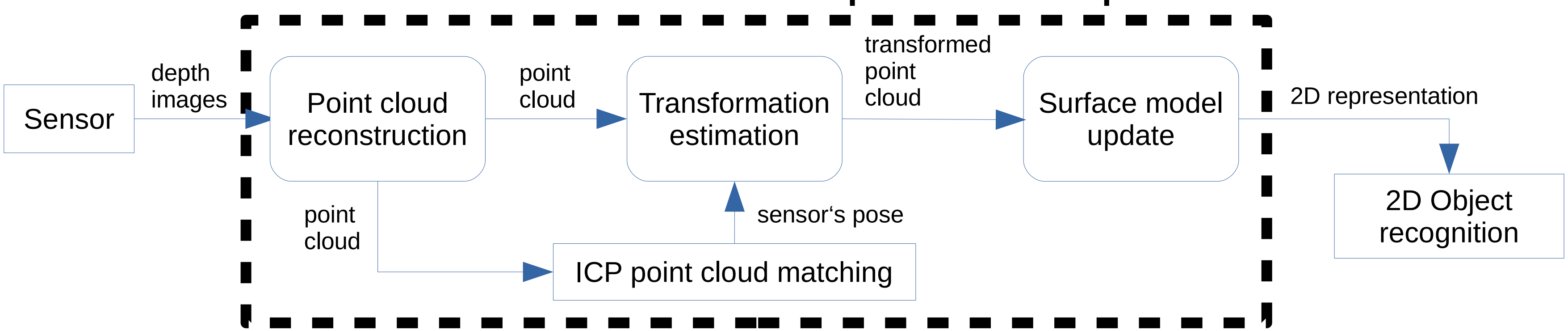
Reconstruction pipeline: black box

Assisted approach:

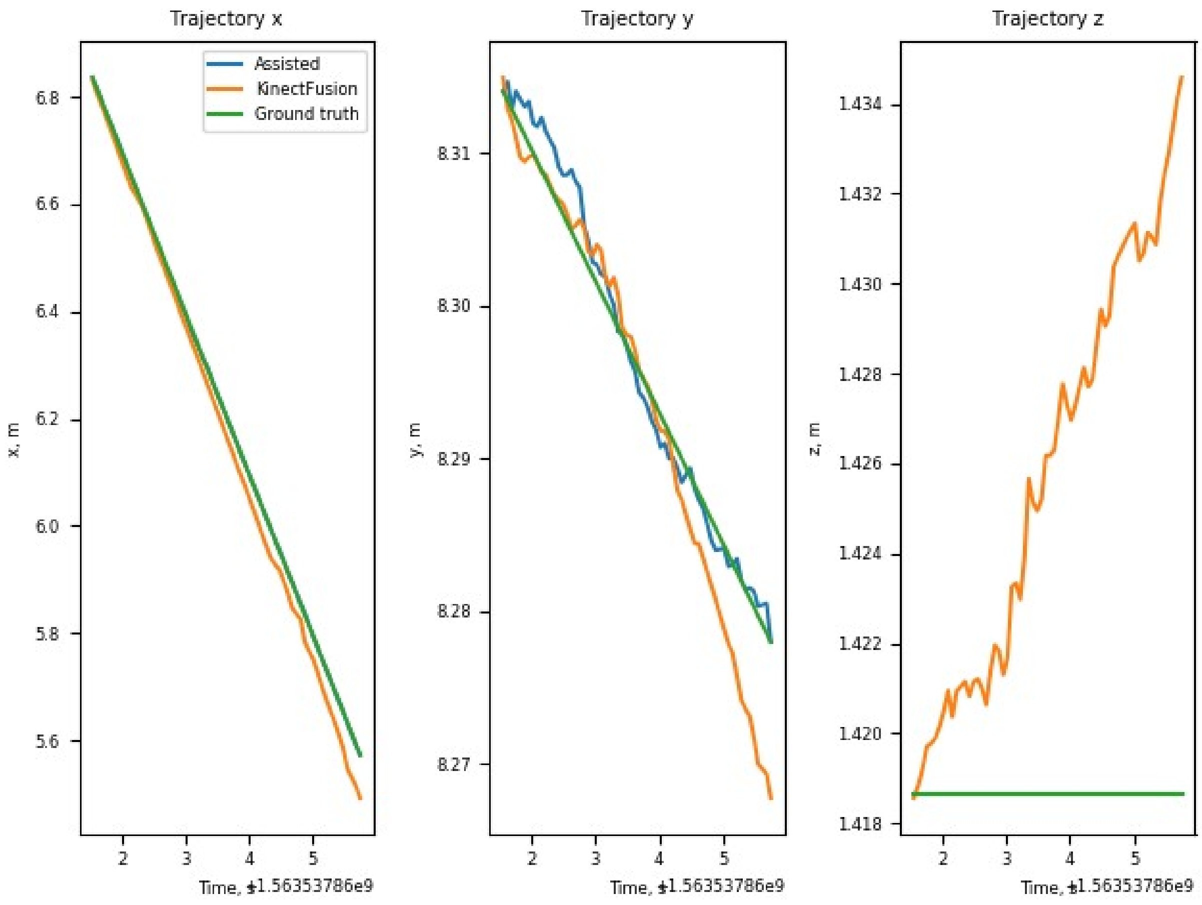


KinectFusion approach:

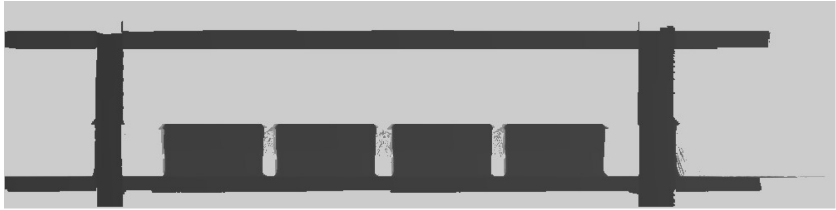
Black box: OpenCV 4.0.1 implementation



Measure of quality: absolute trajectory error (ATE) relative to ground truth trajectories

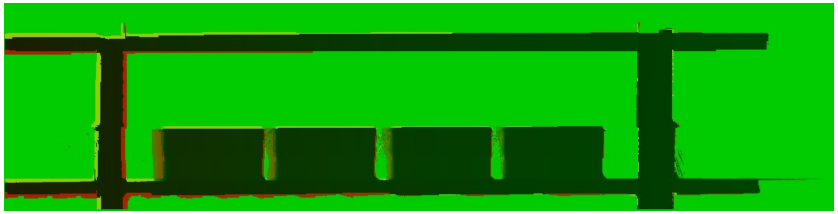


Measure of quality: pixel-wise matching with ground truth images



Ground truth

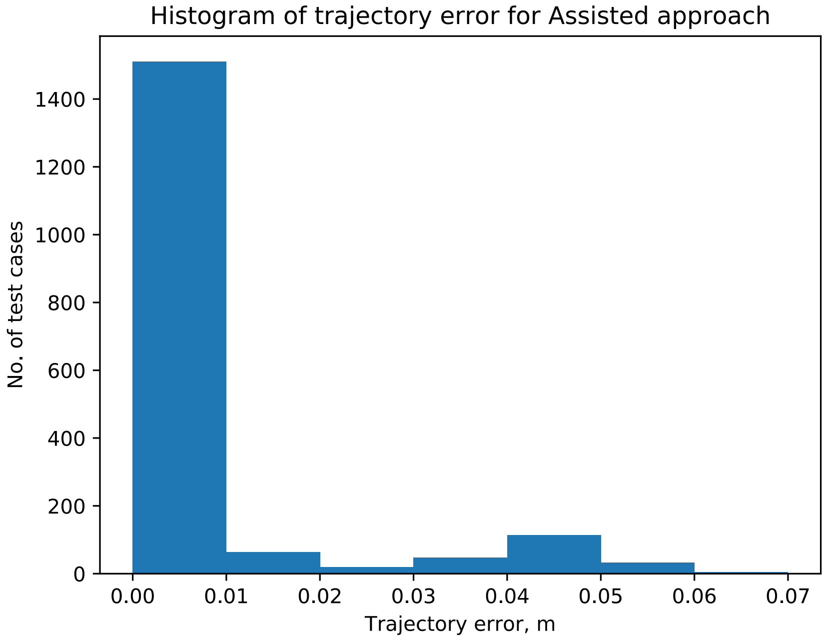
Reconstructed image

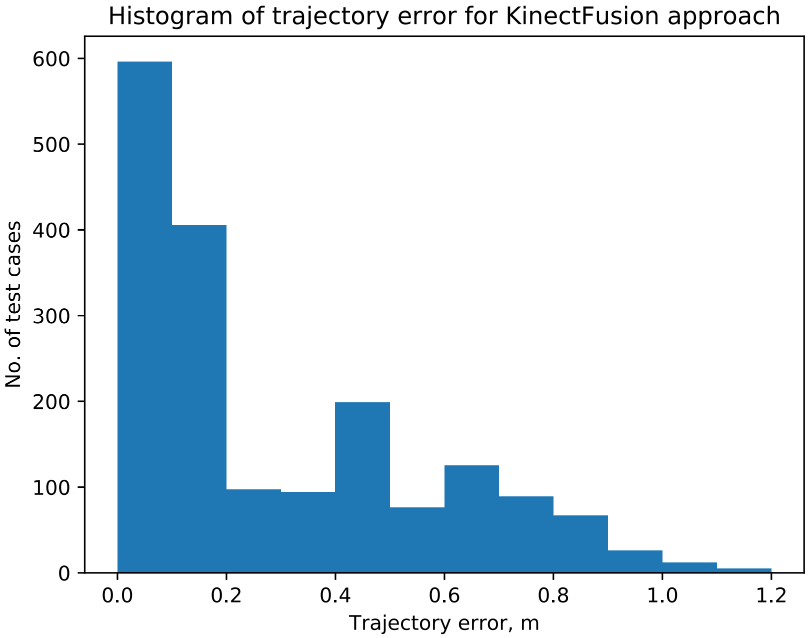


Matched images

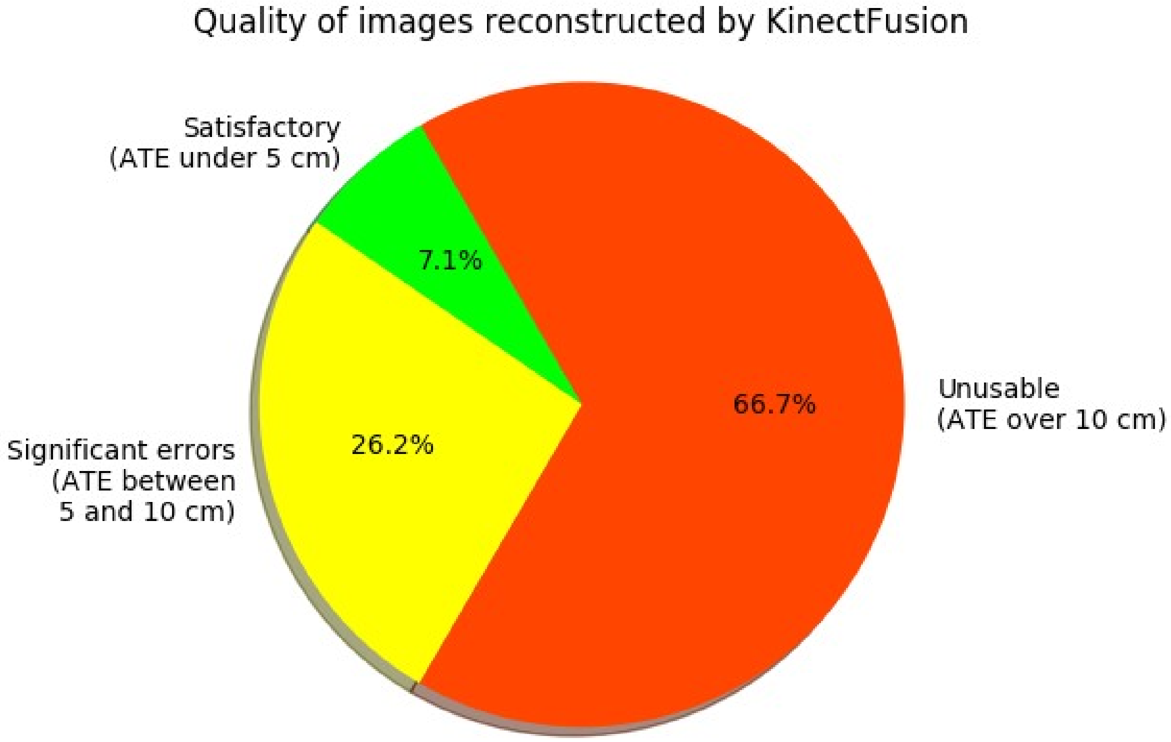
Mismatched pixels

Main result: better performance of the Assisted approach

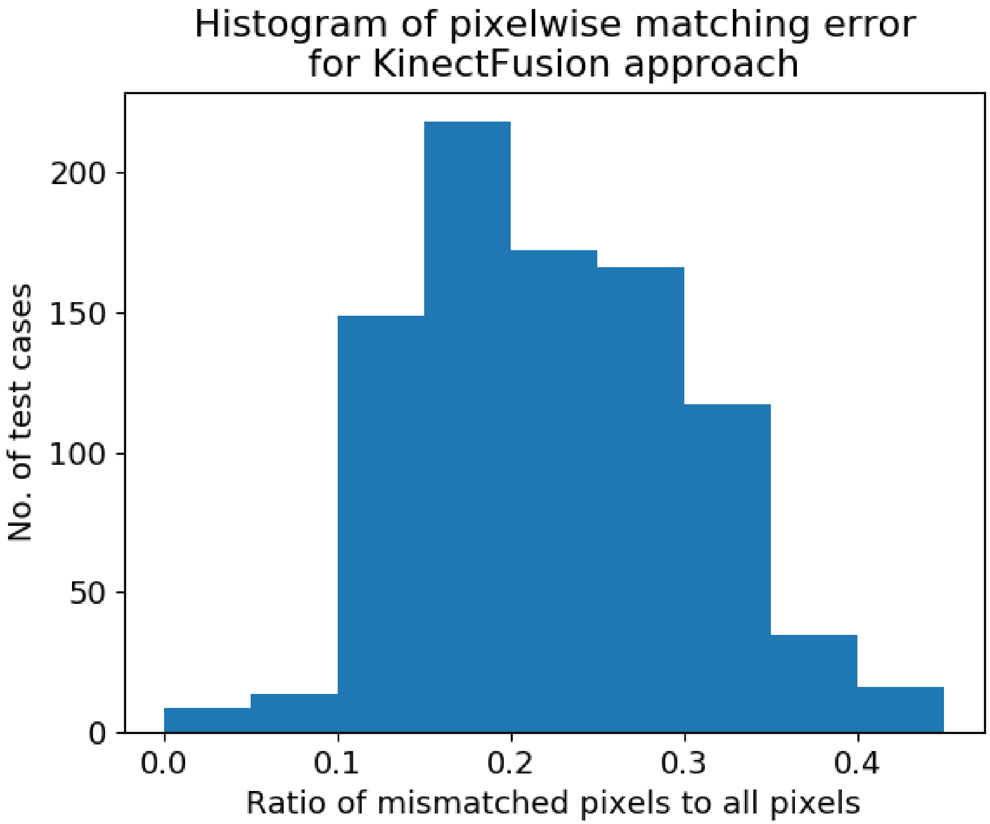




Main result: better performance of the Assisted approach

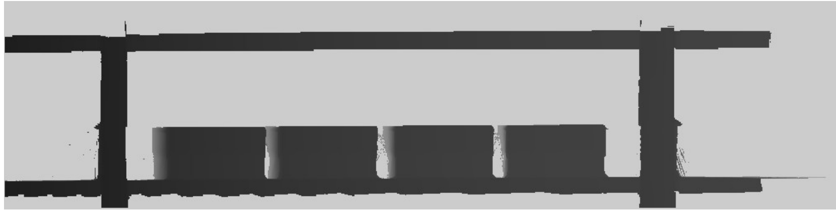


Main result: better performance of the Assisted approach

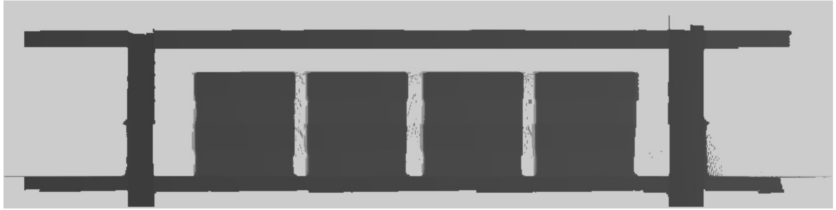


Some of the best KinectFusion results (by pixel mismatch ratio)

Ground truth



Pixel mismatch ratio 4.56%,
ATE 5.35 cm



Ground truth

Pixel mismatch ratio 4.94%,
ATE 5.28 cm

Some of the worst KinectFusion results (by pixel mismatch ratio)

Ground truth

Pixel mismatch ratio 42.8%,
ATE 92.7 cm

Ground truth

Pixel mismatch ratio 34%,
ATE 70.5 cm

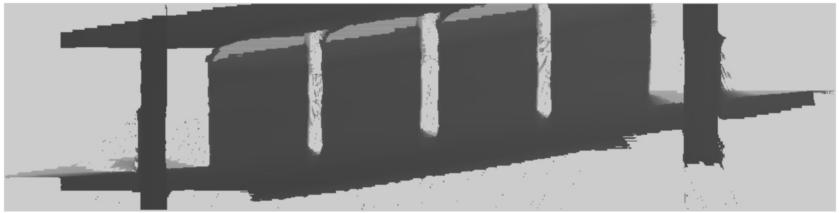
Some of the worst Assisted approach results (by ATE)

ATE 5.97 cm

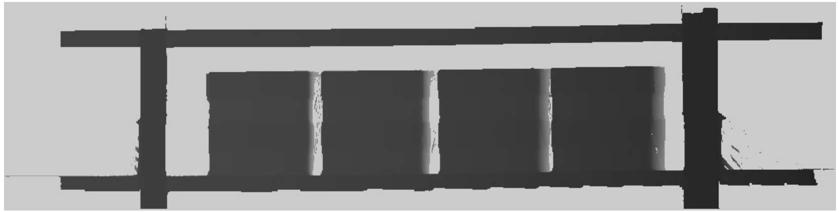
ATE 6.75 cm

KinectFusion’s immunity to external localization error

The reconstruction error in the Assisted approach can be made arbitrarily large by introducing localization error artificially (example below: constant drift in z-direction). In the KinectFusion approach, it remains unaffected.

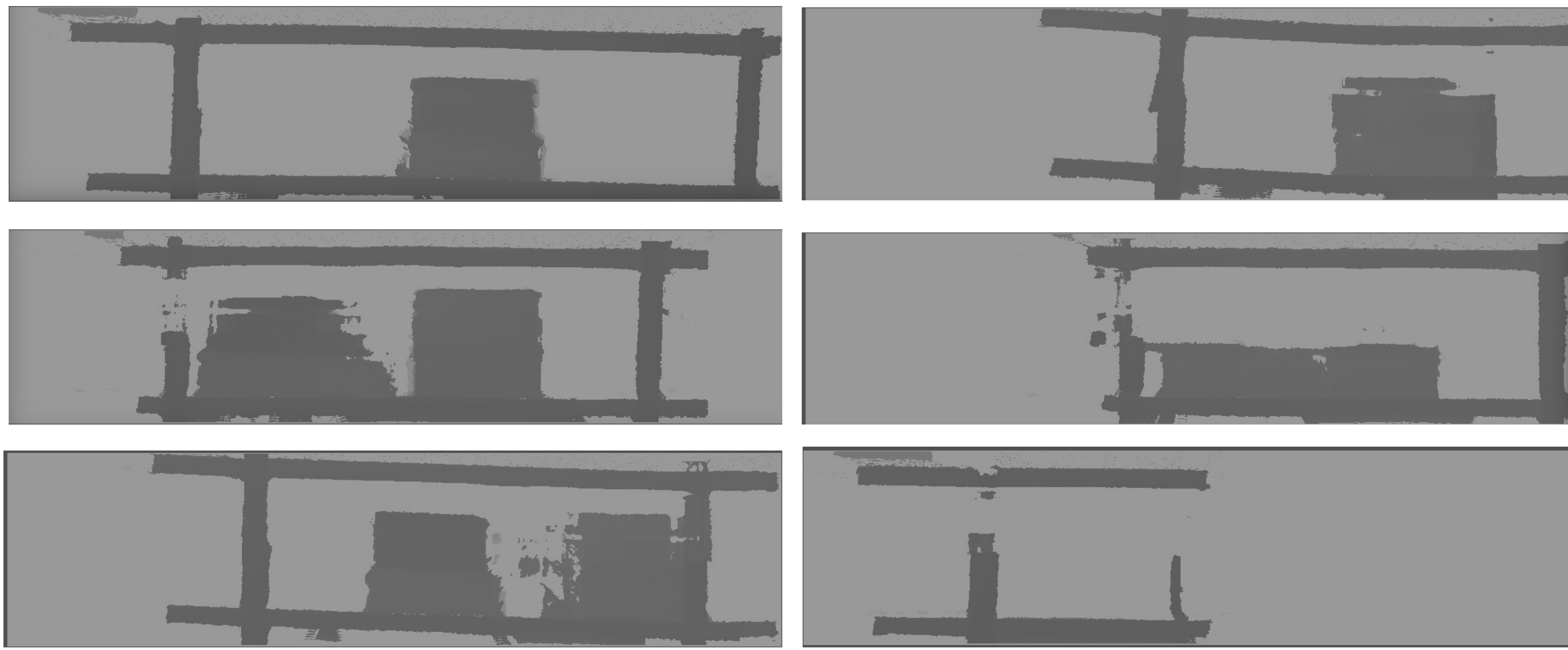


Assisted approach



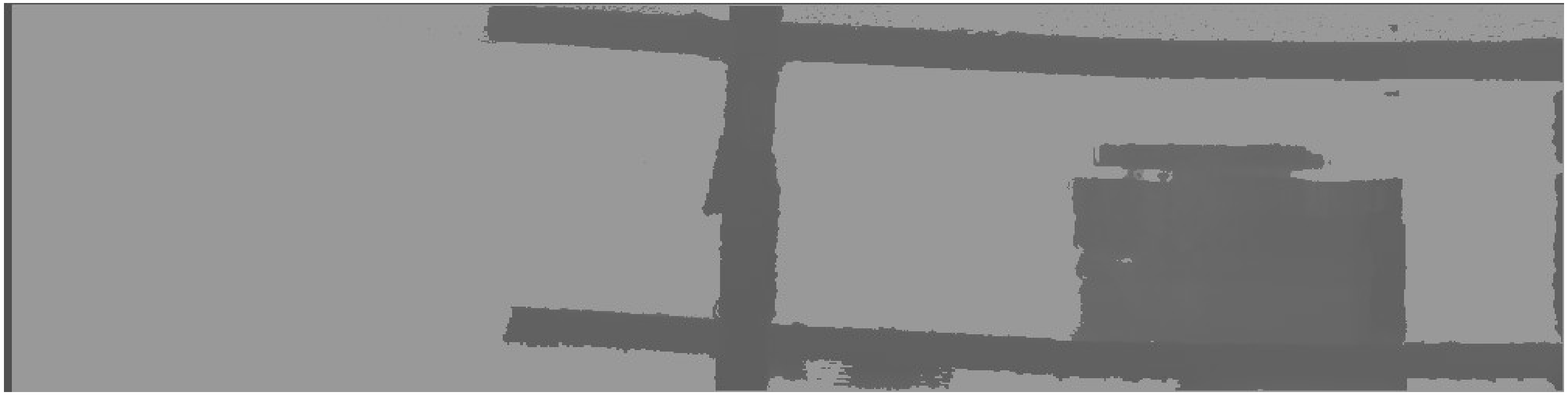
KinectFusion approach

Errors in KinectFusion reconstructions

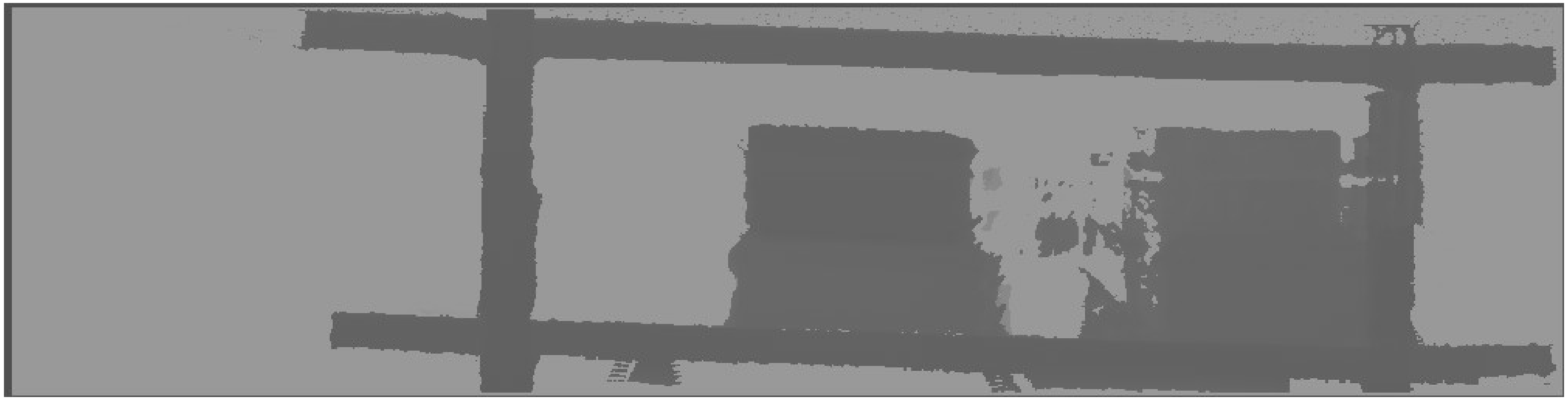


Shortening

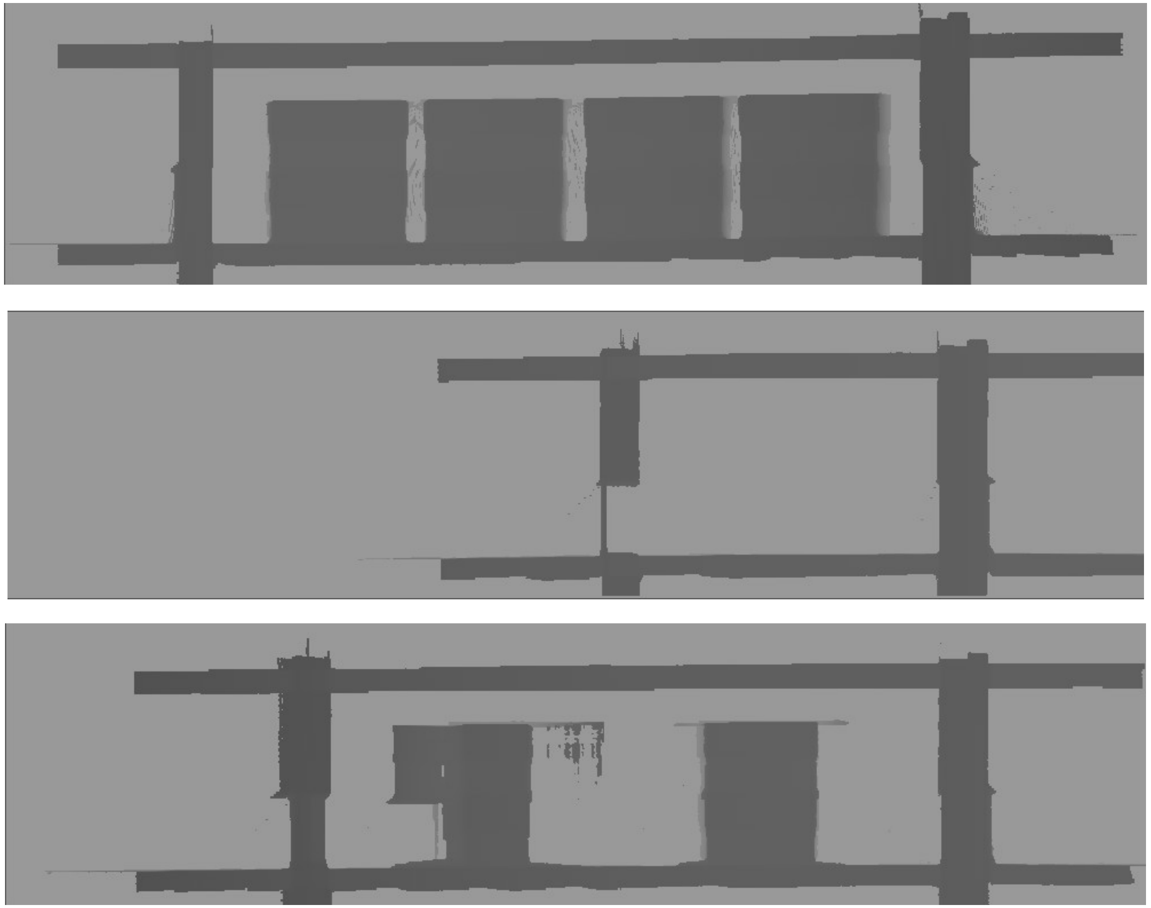
Bending and skewing



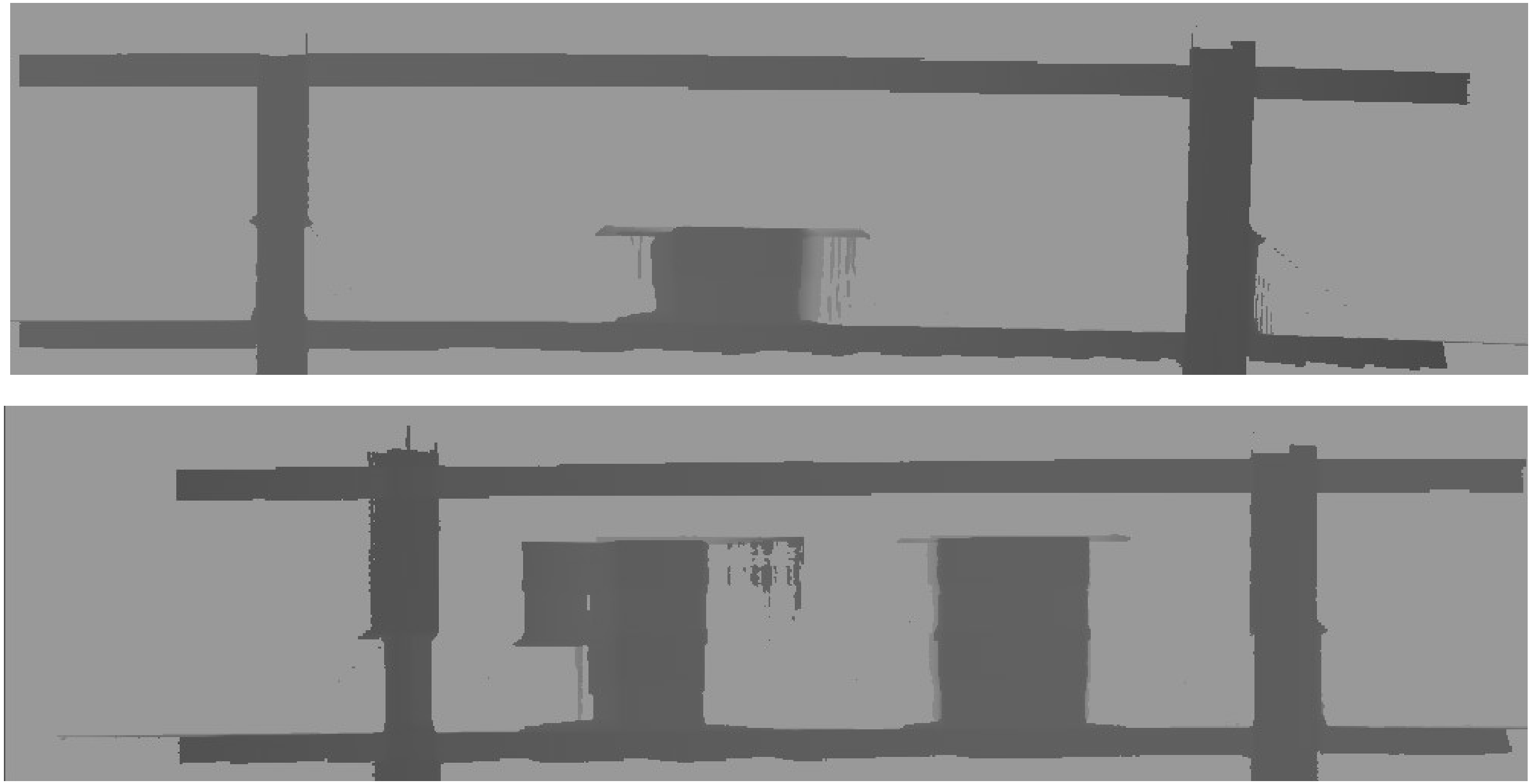
Box shape distortion



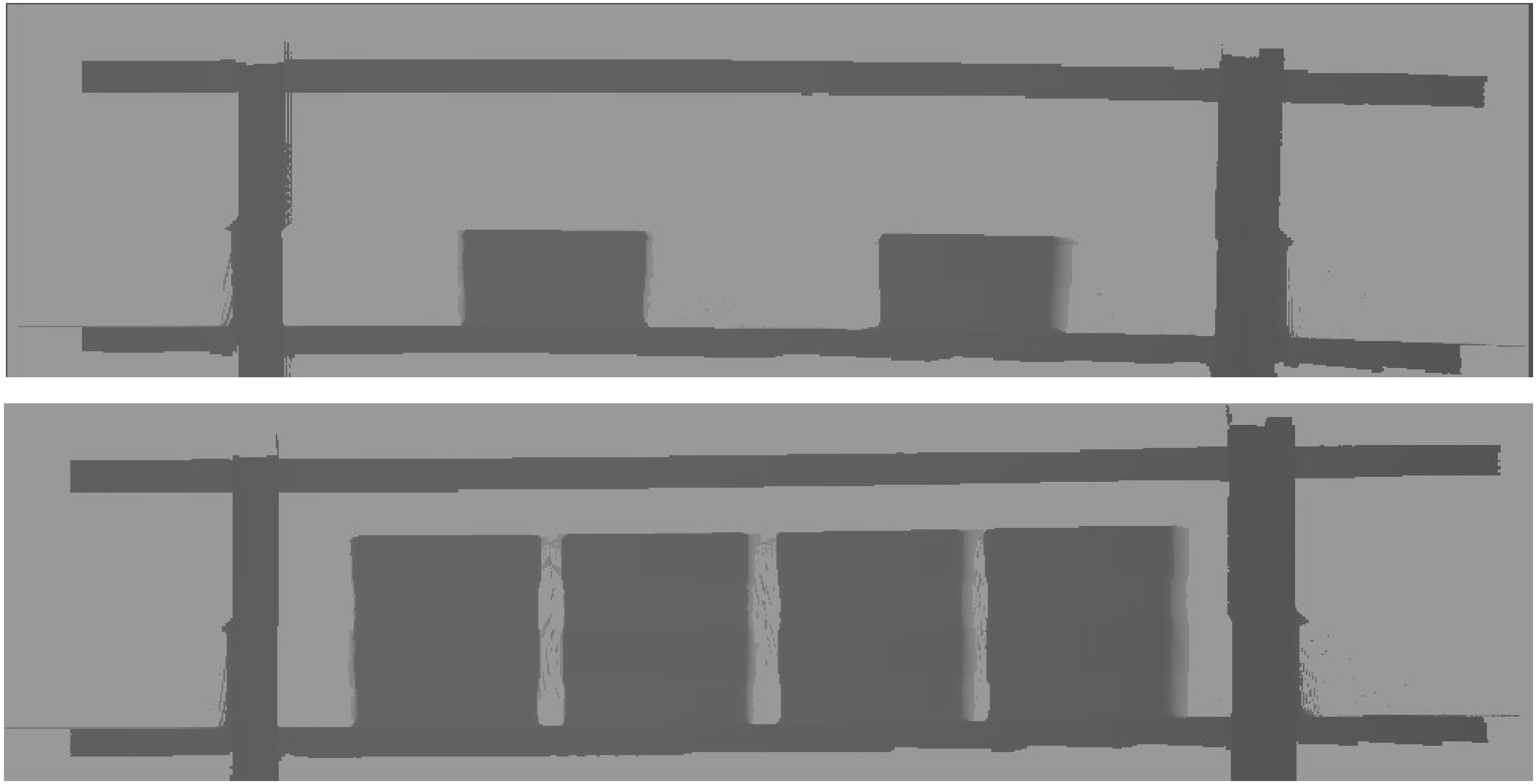
Simulated data: same errors



Simulated data: box shape distortion

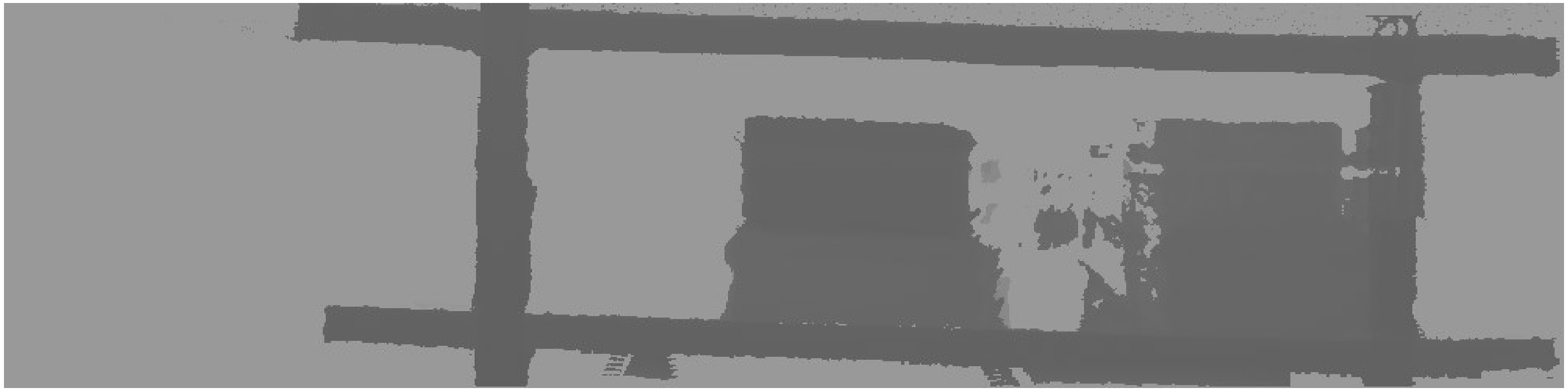


Simulated data: bending

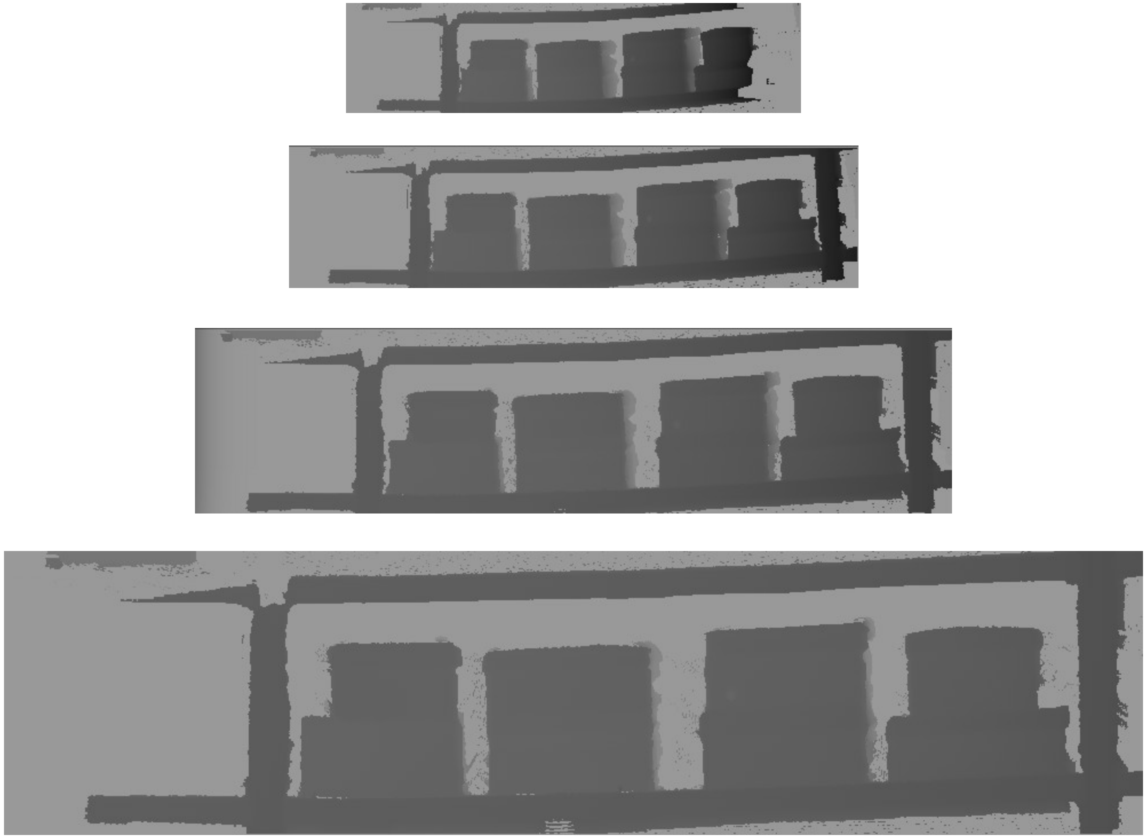


Simulated data: shortening

Multiple errors



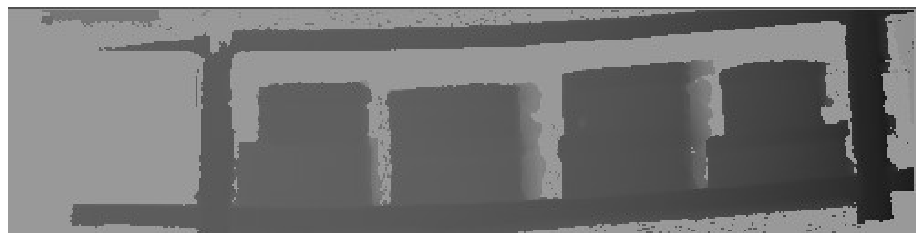
In general: quality grows with resolution



In general: quality grows with resolution

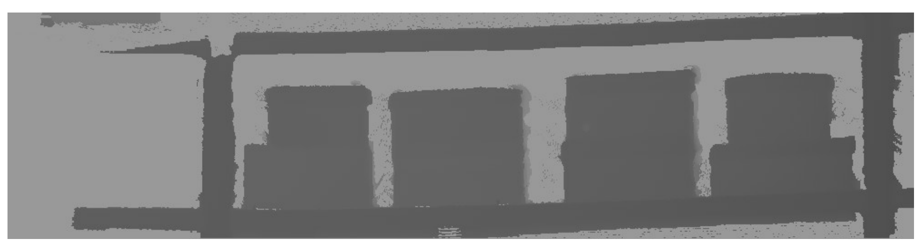
Voxel size: 5mm

4mm



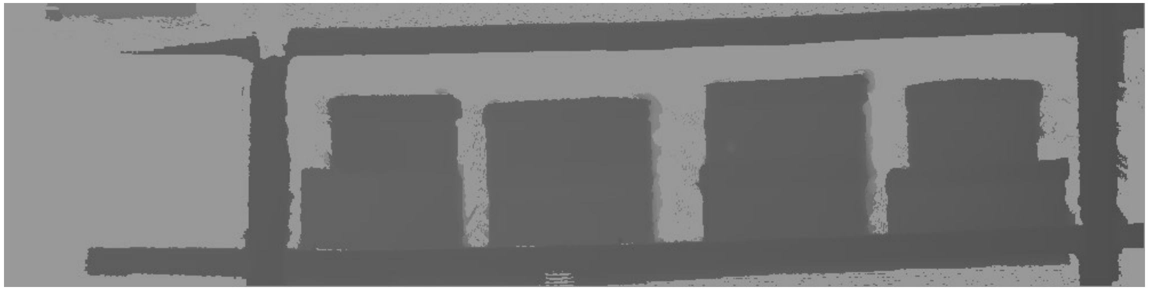
3mm

2mm



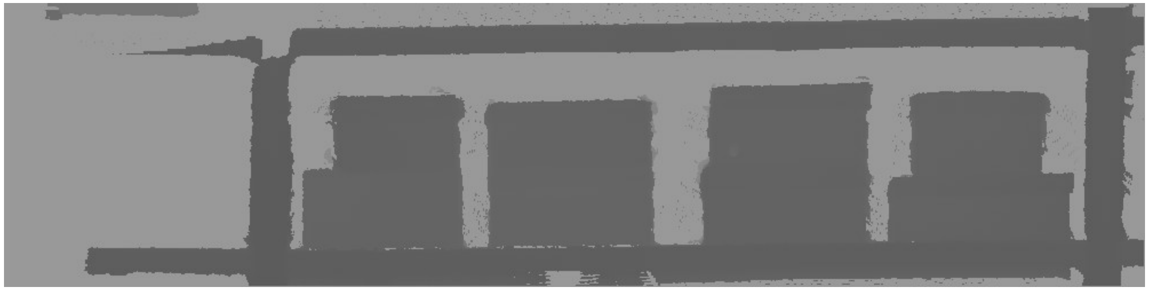
Quality grows with downsampling rate

Using every message:



Using 1 in every 4 messages:

Using 1 in every 8 messages:



Conclusion

The KinectFusion algorithm is not suitable for the examined scene geometry. The research of the KinectFusion approach will be abandoned by Magazino GmbH.

The Assisted approach shows decent performance provided that the localization error does not exceed certain levels and will be further used in production of TORU 5.

Possible further research direction: combining the depth data with the RGB data from RGB-D sensors and using corresponding simultaneous localization and mapping (SLAM) algorithms:

- BAD SLAM
- Voxblox

References

Newcombe, R. A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A. J., Kohli, P., Shotton, J., Hodges, S., Fitzgibbon, A. (2011).
KinectFusion: Real-Time Dense Surface Mapping and Tracking.
Paper presented at The 10th IEEE International Symposium on Mixed and Augmented Reality (ISMAR 2011), Basel, Switzerland.
DOI: 10.1109/ISMAR.2011.6092378

Oleynikova, H., Taylor, Z., Fehr, M., Nieto, J., Siegwart, R. (2017).
Voxblox: Incremental 3D Euclidean Signed Distance Fields for On-Board MAV Planning.
Paper presented at IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, Canada.
DOI: 10.1109/IROS.2017.8202315

Schops, T., Sattler, T, Pollefeys, M. (2019).
BAD SLAM: Bundle Adjusted Direct RGB-D SLAM.
Paper presented at The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 134-144, Long Beach, California, USA.

https://github.com/ETH3D/badslam

https://github.com/ethz-asl/voxblox