Karlsruher Institut für Technologie

mfi21 Karlsruhe

Karlsruhe, Germany
23–25 September 2021

Object Detection and Mapping with Bounding Box Constraints

Benchun Zhou, Aibo Wang, Jan-Felix Klein, Furmans Kai
Institut for Material Handling and Logistics (IFL)
Karlsruhe Institute of Technology (KIT)
Karlsruhe, Germany

Institut für Fördertechnik und Logistiksysteme (IFL)

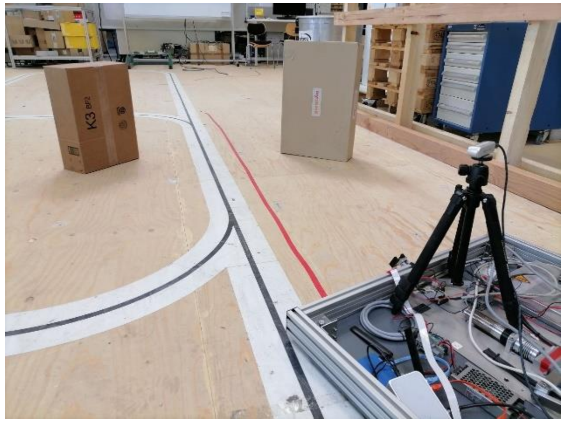
Context

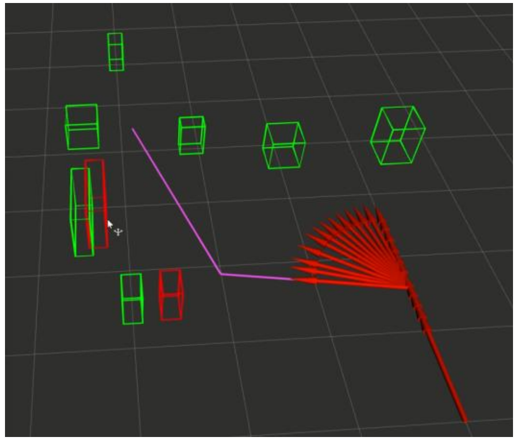
- Motivation
- Problems
- Methods
- Results
- Conclusions

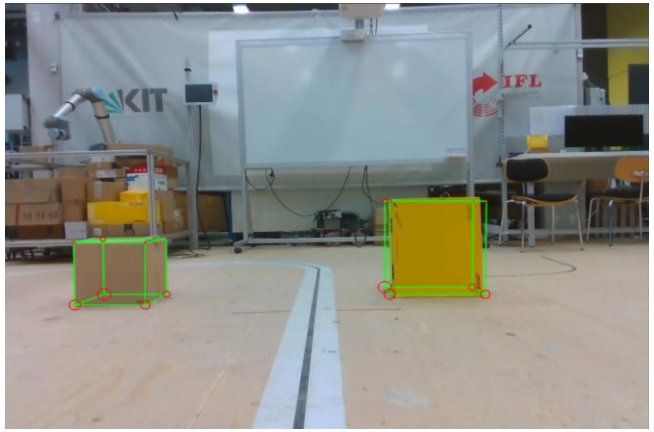
Movitation

- **Intralogistic environment**: most objects are standard cuboid (parcels, contrainers, …)
- Objects provide useful information for logistic tasks: navigation, grasping, …

- **Sensor Fusion Systems** (LiDAR, Camera, IMU, …): stable, complex, higher cost.
- **Camera-based Systems** (Monocular, Stereo, Depth): flexible, cheap, not robust





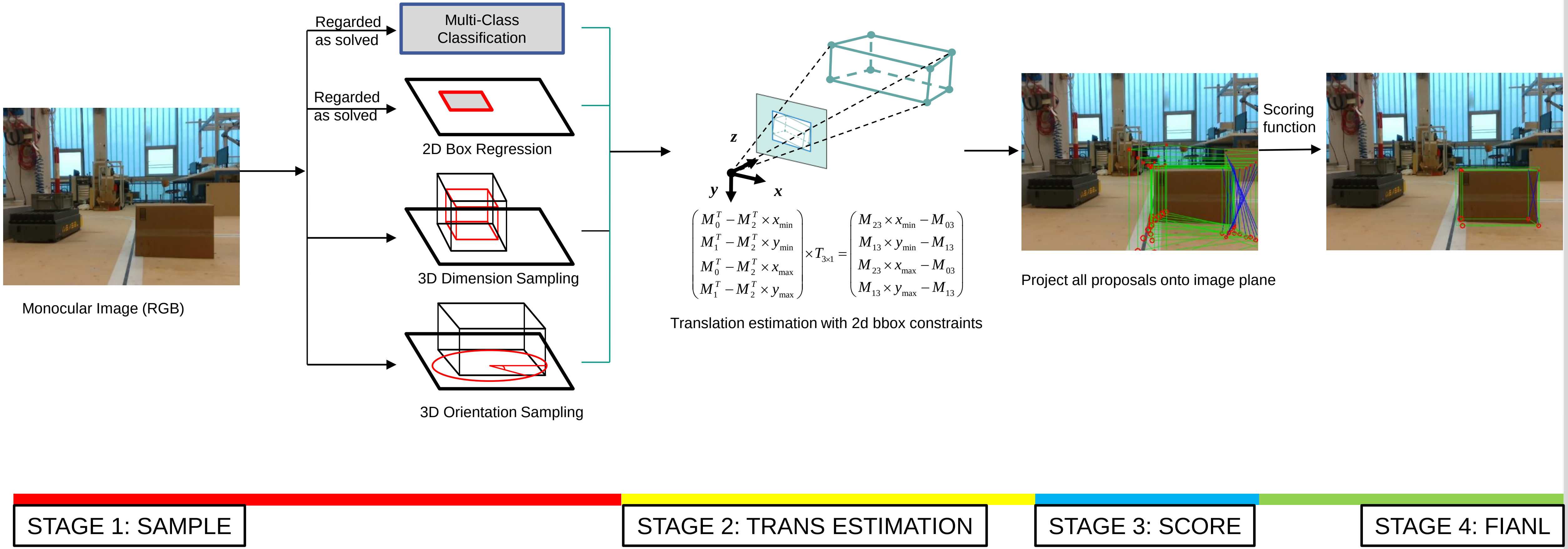


Problems

- **1, 3D object detection from single image (cuboid format)**
  - Use only RGB images
  - Use geometry features and constraints
- **2, 3D object based localization and mapping**
  - Jointly optimize objects and robot location
  - Build a general map with points and objects

Methods: 3D object detection from single image

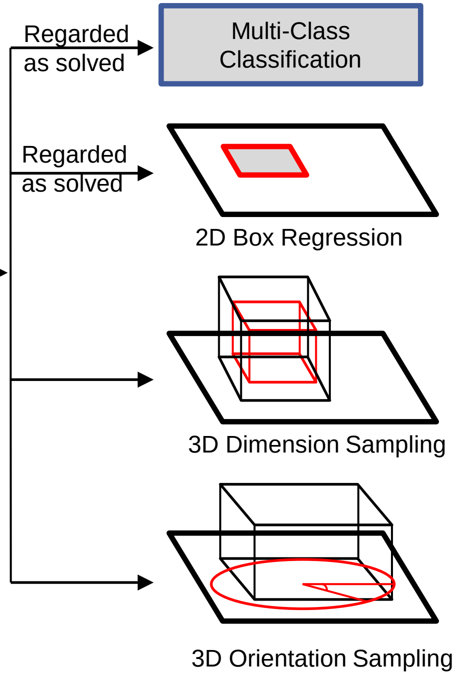
- Framework of proposed method:



Methods: 3D object detection from single image

- Stage 1: Sample

Monocular Image (RGB)



YOLOv4, SSD, faster R-CNN

YOLOv4, SSD, faster R-CNN

Dimension based on class name
Parcel: 40x30x20 cm
Chair: 50x50x50 cm
……

For objects lay in the ground:
Roll: 0
Ptich: 0
Yaw: sample from [0-pi]

- Input:
  - One RGB image (monocular)
- Output:
  - ✓ Classname
  - ✓ 2D bounding box
  - ✓ Cuboid proposals (dimension and orientation pair)

Methods: 3D object detection from single image

- Stage 2: Translation estimation with 2D bounding box constraints

For perspetive camera, we have projection function as:

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = K_{3\times 3}\left[R'_{camera} \quad T'_{camera}\right]\begin{pmatrix} X_{3D} \\ 1 \end{pmatrix}$$

Points in image

Camera info: calib, rotation, translation

Points in 3d space

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \mathrm{P}_{rojection}\begin{pmatrix} X_{3D} \\ 1 \end{pmatrix} = \mathrm{P}_{rojection}\begin{pmatrix} I_{3\times 3} & R_{3\times 3}(\theta)\cdot D_{3\times 1} \\ 0 & 1 \end{pmatrix}\begin{pmatrix} T_{3\times 1} \\ 1 \end{pmatrix}$$

Cuboid corners in image

Camera info

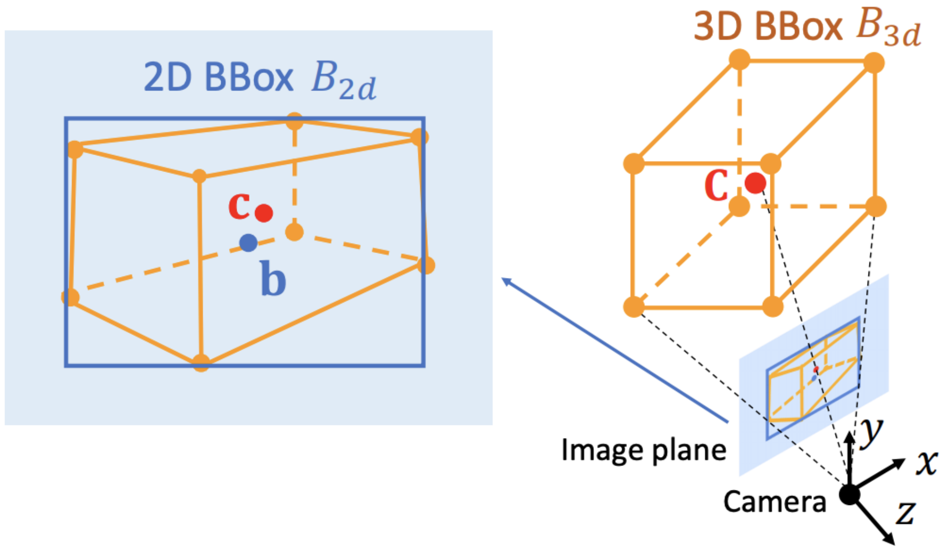
Cuboids info: rotation, dimension, translation

Corners should fit 2D bounding box in image, known

known

rotation and dimension: sampled from stage 1, known

unknown



Methods: 3D object detection from single image

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = P_{rojection} \begin{pmatrix} X_{3D} \\ 1 \end{pmatrix} = P_{rojection} \begin{pmatrix} I_{3\times3} & R_{3\times3}(\theta)\cdot D_{3\times1} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} T_{3\times1} \\ 1 \end{pmatrix}$$

For right constraits of bounding box, we get one function with 3 variables.

$$x_{max} = \left(M_{3\times4} \begin{pmatrix} T_{3\times1} \\ 1 \end{pmatrix} \right)_x = \left(\begin{pmatrix} M[0,0:3] & M[0,3] \\ M[1,0:3] & M[1,3] \\ M[2,0:3] & M[2,3] \end{pmatrix} \begin{pmatrix} T_{3\times1} \\ 1 \end{pmatrix} \right)_x$$

$$= \left(\begin{pmatrix} M_0^T & M_{03} \\ M_1^T & M_{13} \\ M_2^T & M_{23} \end{pmatrix} \begin{pmatrix} T_{3\times1} \\ 1 \end{pmatrix} \right)_x = \begin{pmatrix} M_0^T \times T_{3\times1} + M_{03} \\ M_!^T \times T_{3\times1} + M_{13} \\ M_2^T \times T_{3\times1} + M_{23} \end{pmatrix}_x$$

$$= \frac{M_0^T \times T_{3\times1} + M_{03}}{M_2^T \times T_{3\times1} + M_{23}}$$

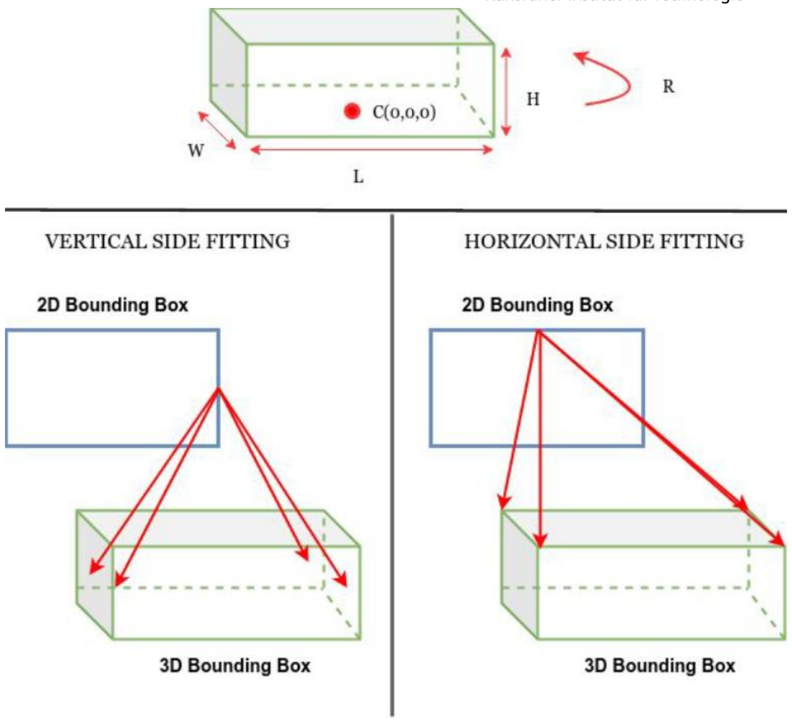
$$\left(M_0^T - M_2^T \times x_{max} \right) \times T_{3\times1} = M_{23} \times x_{max} - M_{03}$$

Take all the 2D bounding box constraints into consideration

$$\begin{pmatrix} M_0^T - M_2^T \times x_{min} \\ M_1^T - M_2^T \times y_{min} \\ M_0^T - M_2^T \times x_{max} \\ M_1^T - M_2^T \times y_{max} \end{pmatrix} \times T_{3\times1} = \begin{pmatrix} M_{23} \times x_{min} - M_{03} \\ M_{13} \times y_{min} - M_{13} \\ M_{23} \times x_{max} - M_{03} \\ M_{13} \times y_{max} - M_{13} \end{pmatrix}$$

Over-constrained system, solved by least squares method:

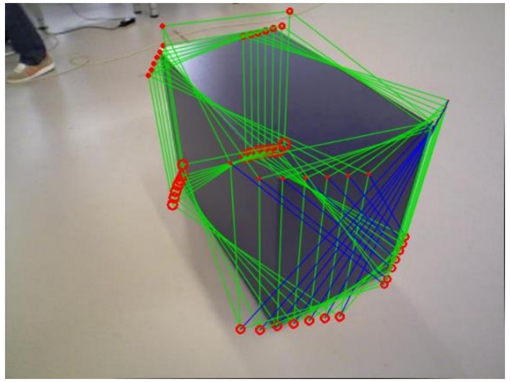
$$A \times T_{3\times1} = b\,(b \neq 0) \rightarrow T_{3\times1} = \left(A^T A\right)^{-1} A^T b$$

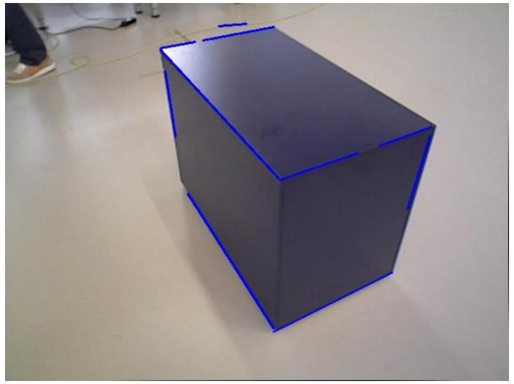


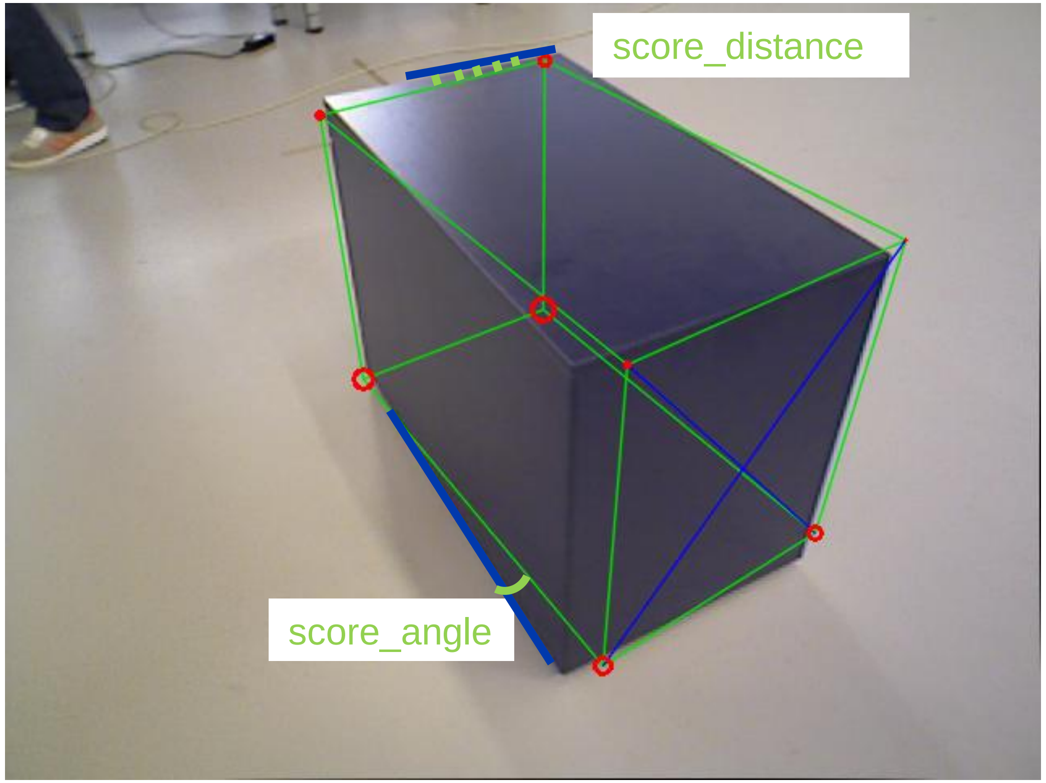
- Input:
  - 2D bbox, camera info
  - Cuboid proposals (dimension, orientation)
- Output:
  - ✓ Object info in 3D Space (translation, dimension, orientaion)

Methods: 3D object detection from single image

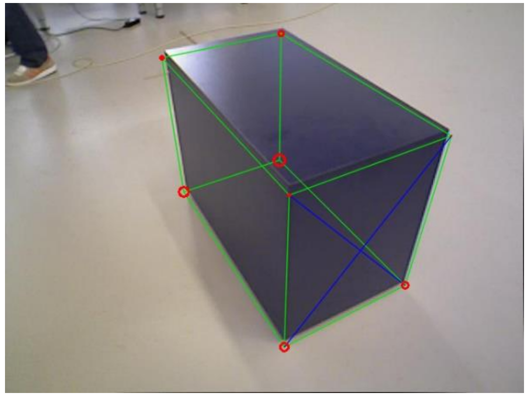
- Stage 3: score every cuboid proposals with detected lines







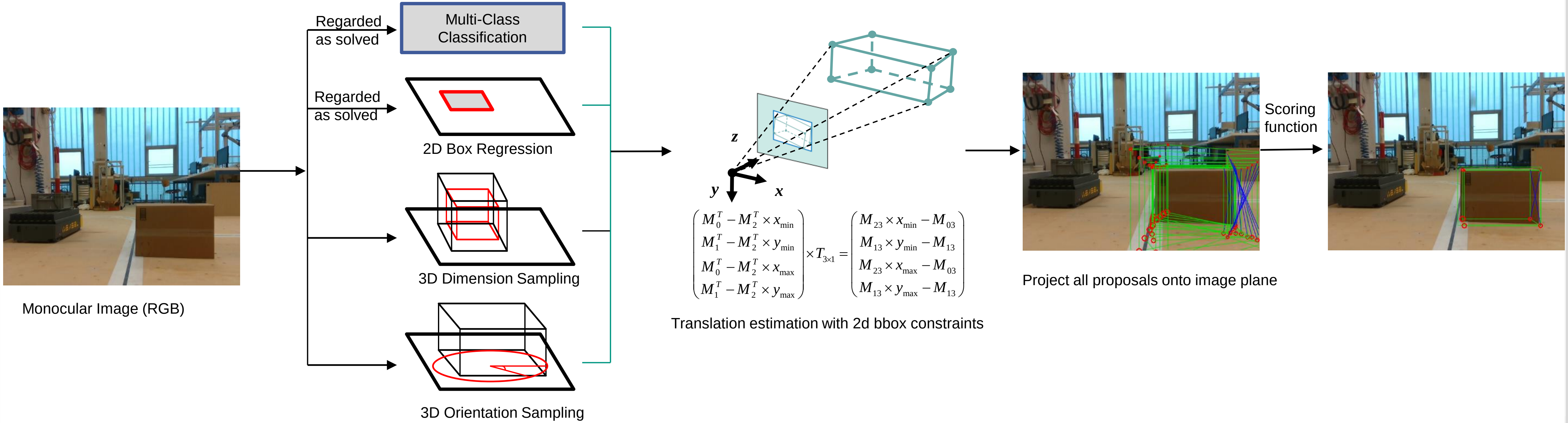
- Input:
  - Cuboid propoals in 3D space
- Output:
  - ✓ Cuboid proposals in image
  - ✓ Scores (based on edge)
  - ✓ One best proposals that fits object.



cost function $$E(\underbrace{O}_{\text{object}} \mid \underbrace{I}_{\text{edges}}) = \omega_1 \underbrace{\phi_{\text{bbox}}(O \mid I)}_{\text{Bbox cost}} + \omega_2 \underbrace{\phi_{\text{angle}}(O \mid I)}_{\text{angle cost}} + \omega_3 \underbrace{\phi_{\text{dist}}(O \mid I)}_{\text{distance cost}}$$

Methods: 3D object detection from single image

- Framework of proposed method:



STAGE 1: SAMPLE | STAGE 2: TRANS ESTIMATION | STAGE 3: SCORE | STAGE 4: FIANL

Methods: 3D object detection from single image

- Result: KITTI 3D Object Dataset

- Summarize
  - Images: 1008
  - 2D Bbox: From Yolov3
  - Dimension: average from label
  - Degree sample: every 5 degree in [0, 180]
  - Evaluation: 3D IoU

Method	KITTI Dataset
Deep3DBox[1]	0.33
Mono3D[2]	0.22
CubeSLAM[3]	0.21
Ours*	0.33

Result on Object 3D IoU

Conclusion:
-> Comparable to deep3dbox, and outperform CubeSLAM detection
-> share similar idea with deep3dbox, but do not use deep learning

Methods: 3D object detection from single image

- Result: SUN RGBD Dataset and Logistic Dataset

Result on Object 3D IoU

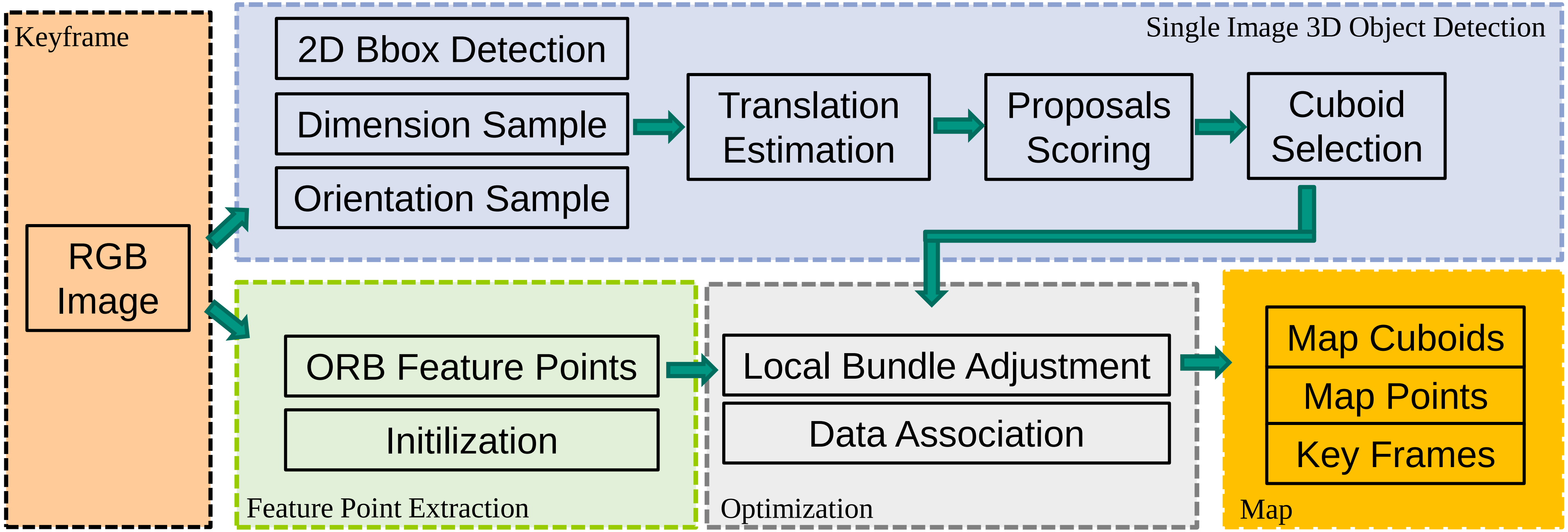
Method	SUN RGBD Dataset	Logistic Dataset
CubeSLAM[3]	**0.39**	0.32
Ours*	0.31	**0.41**

- Summarize
  - Images: 200
  - 2D Bbox: From Yolov3
  - Dimension: average from label or mesurement
  - Degree sample: every 5 degree in [0, 180]
  - Evaluation: 3D IoU

Conclusion:
-> proposed method works best for boxy like objects.
-> background influences the score function.
-> camera angle influences the sample number.

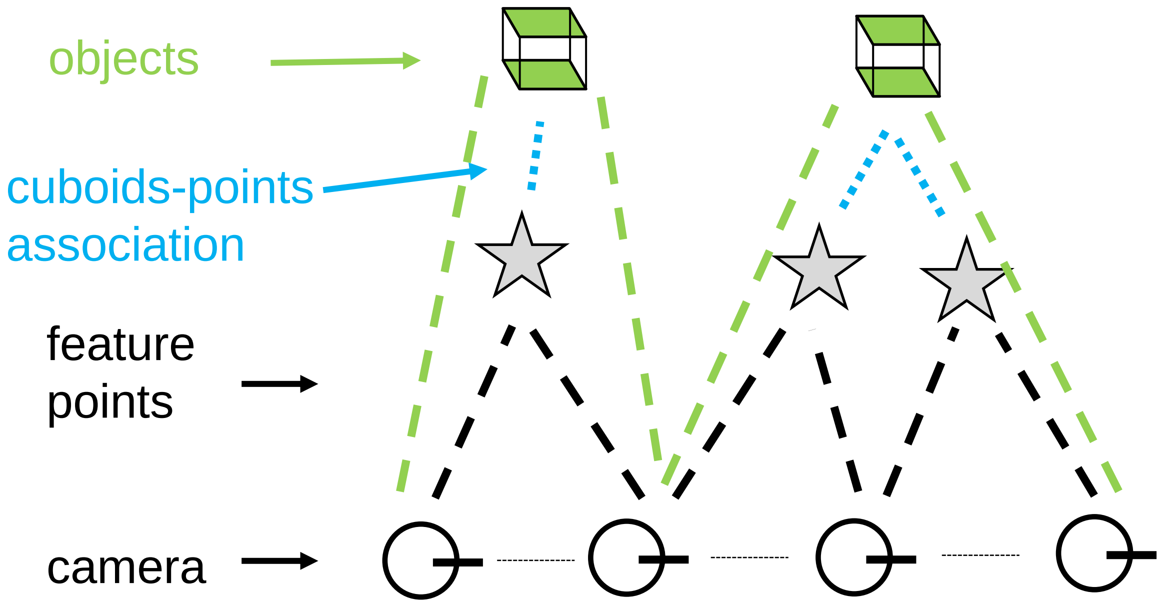
Methods: 3D object-based localization and mapping

- Framework: Use feature points and cuboids as landmarks to realize localization and mapping



Methods: 3D object-based localization and mapping

- Framework: use feature points and cuboids as landmarks to realize localization and mapping



$$C^*, O^*, P^* = \underset{\{C,O,P\}}{\arg\min} \sum_{C_i, O_j, P_k} \left\| e\left(c_i, o_j\right) \right\|^2_{\Sigma_{ij}} + \left\| e\left(c_i, p_k\right) \right\|^2_{\Sigma_{ik}} + \left\| e\left(o_j, p_k\right) \right\|^2_{\Sigma_{jk}}$$

Camera-objects measurement | Camera-points measurement | Object-points measurement

- Camera-point measurement (from ORB-SLAM2 [4]):
  - reprojection error between matched 3D points P in world coordinates and keypoints

$$e\left(c_i, o_j\right) = \pi\left(T_c^{-1} P\right) - z_m$$

- Camera-object measurement
  - reprojection error between matched 3D object corners in world coordinates and detected object corners

$$e\left(c_i, o_j\right) = \pi\left(T_c^{-1} O\right) - y_m$$

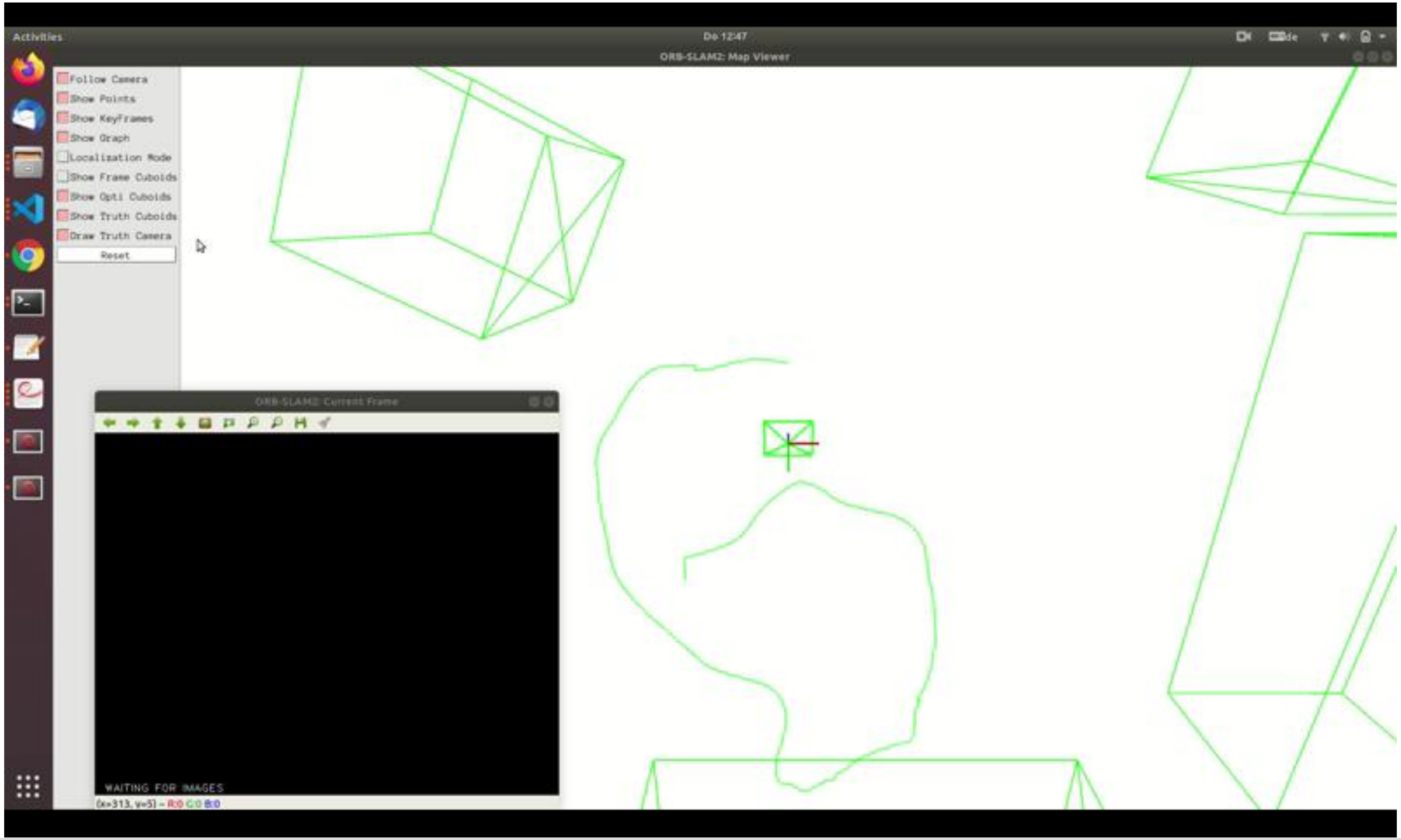
- Object-point measurement
  - distance error between cuboid dimension and object-point distance

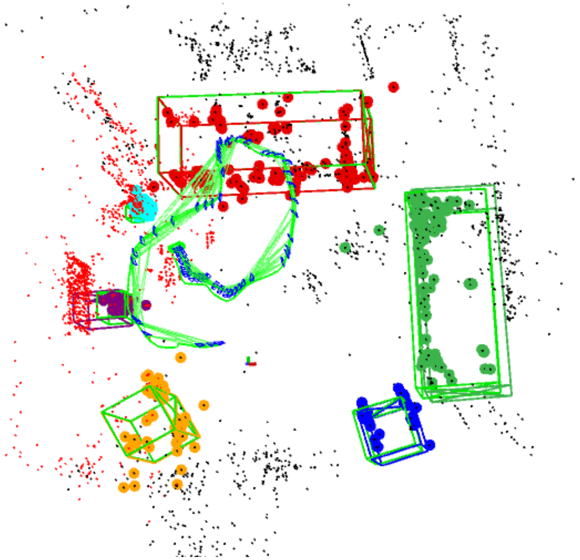
$$e\left(o_j, p_k\right) = \max\left(\left|T_o^{-1} P\right| - \mathbf{d}_m, \mathbf{0}\right)$$

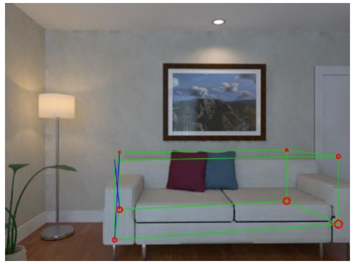
More info, please see our paper

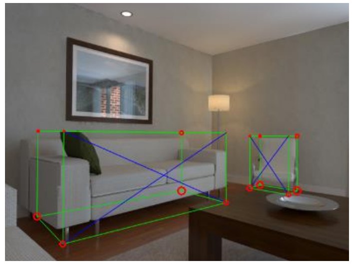
Methods: 3D object-based localization and mapping

- Result: ICL-NUIM Dataset (living room)









RMSE on TUM ICL NUIM Dataset

Method	ICL NUIM
CubeSLAM[3]	**0.03**
Ours*	**0.03**

Conclusions

Karlsruher Institut für Technologie

- **Problem:**
  - **3D object detection from single image (cuboid format)**
    - Sample + estimate translation + score
    - Work best for boxy like objects
  - **3D object based localization and mapping**
    - Add camera-object, object-points measurement on ORB-SLAM 2
    - Optimize camera localization and build a general map with points and objects
- **Future work:**
  - Object detection on irregular objects, such as cups, lamps, …
  - Another way to sample the dimension of object
  - Use point-object map for navigation or grasping

Reference

- [1] Mousavian, Arsalan, et al. "3D bounding box estimation using deep learning and geometry." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017.
- [2] Chen, Xiaozhi, et al. "Monocular 3d object detection for autonomous driving." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016.
- [3] Yang, Shichao, and Sebastian Scherer. "CubeSLAM: Monocular 3D object SLAM." IEEE Transactions on Robotics 35.4 (2019): 925-938.
- [4] Mur-Artal, Raul, and Juan D. Tardós. "Orb-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras." IEEE Transactions on Robotics 33.5 (2017): 1255-1262.

Thanks for your listening

Karlsruher Institut für Technologie (KIT)
Institut für Fördertechnik und Logistiksysteme (IFL)

Besucheranschrift	Gotthard-Franz-Str. 8 Geb. 50.38 76131 Karlsruhe
Telefon	+49 (721) 608-48600
Telefax	+49 (721) 608-48609
E-Mail	info@ifl.kit.edu
Homepage	http://www.ifl.kit.edu