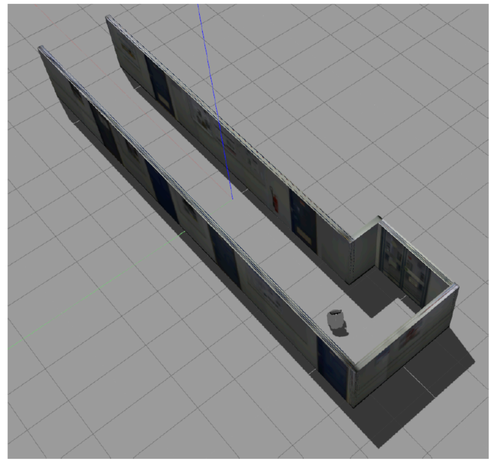
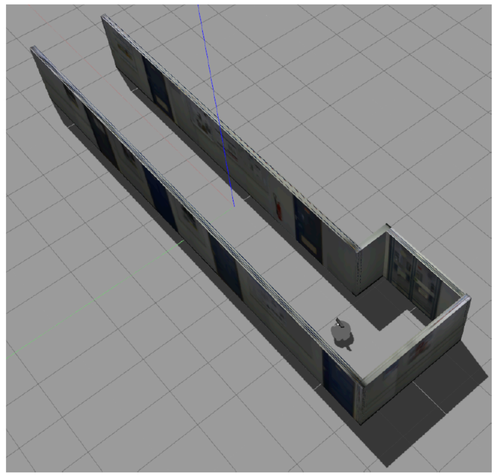
COMPUTER VISION
RESEARCH GROUP

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG





MASTER PROJECT

Autonomous Robot Patrol System

Fabian Behrendt,
Chris Carstensen,
Christian Reichel,
Nicolas Perez de Olaguer,
Danu Caus,
Nana Baah

28.10.2019

Hypothesis

Is it possible, using computer vision techniques, to develop an autonomous robot that surveils a known environment?

Task Description

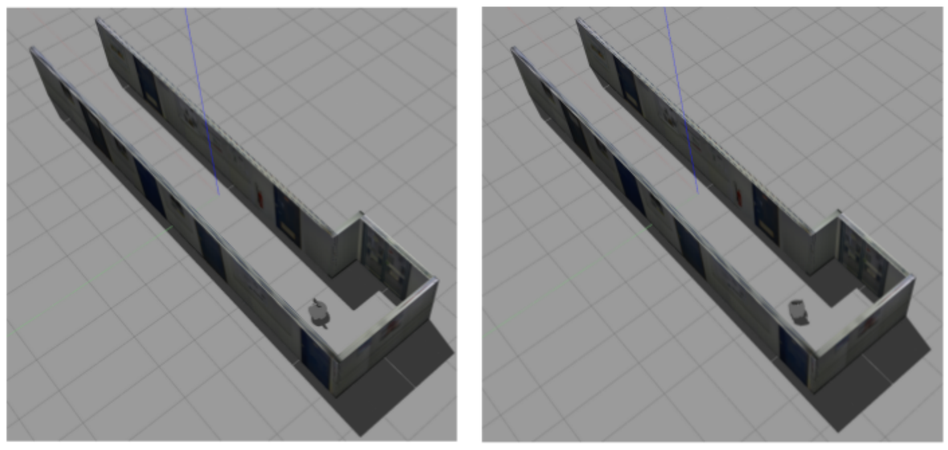
The robot navigates through one of the floors within the University of Hamburg, without prior information, and semantically labels doors as open or closed. It also sends useful information to a human guard in real time via the network.

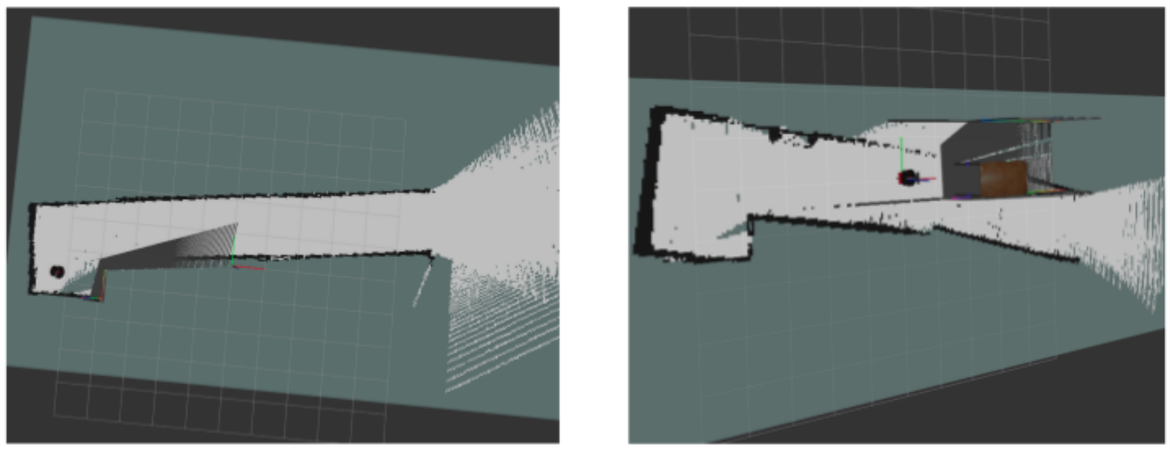
Agenda

1. Background
2. Hardware
3. Architecture
4. Navigation
5. Door Classifier
6. Door Positions
7. Reporter
8. Live Demonstration
9. Conclusion & Discussion

Task Simulation

Prior simulations and predictions





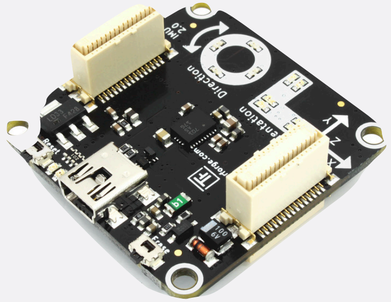
- Drifting maps were indicative of future problems
- Potential solutions: non-CV helper methods

Hardware

Hardware

Pioneer P2 DX Robot

Kinect Camera (v2)



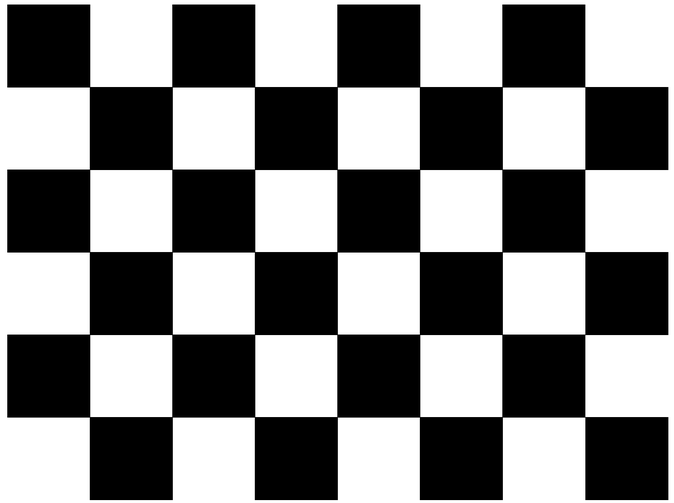
TinkerForge IMU Brick 2.0

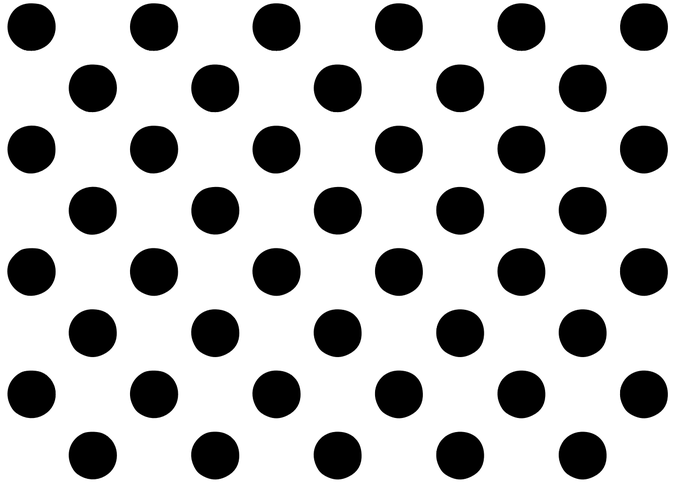
XMG Laptop

Camera calibration

Pattern-based calibration of RGB and infrared camera

- Calibration algorithm, included in IAI Kinect project, using OpenCV
- RGB camera: 6 x 8 checkerboard pattern, 2.6 cm grid size
- Infrared camera: 11 x 4 asymmetric circle pattern
- Captured 191 color, 399 infrared and 125 synced (RGB + infrared) images to calibrate the Kinect 2 camera





Algorithm was not able to correctly recognize the edges of checkerboard. Image was too blurry.

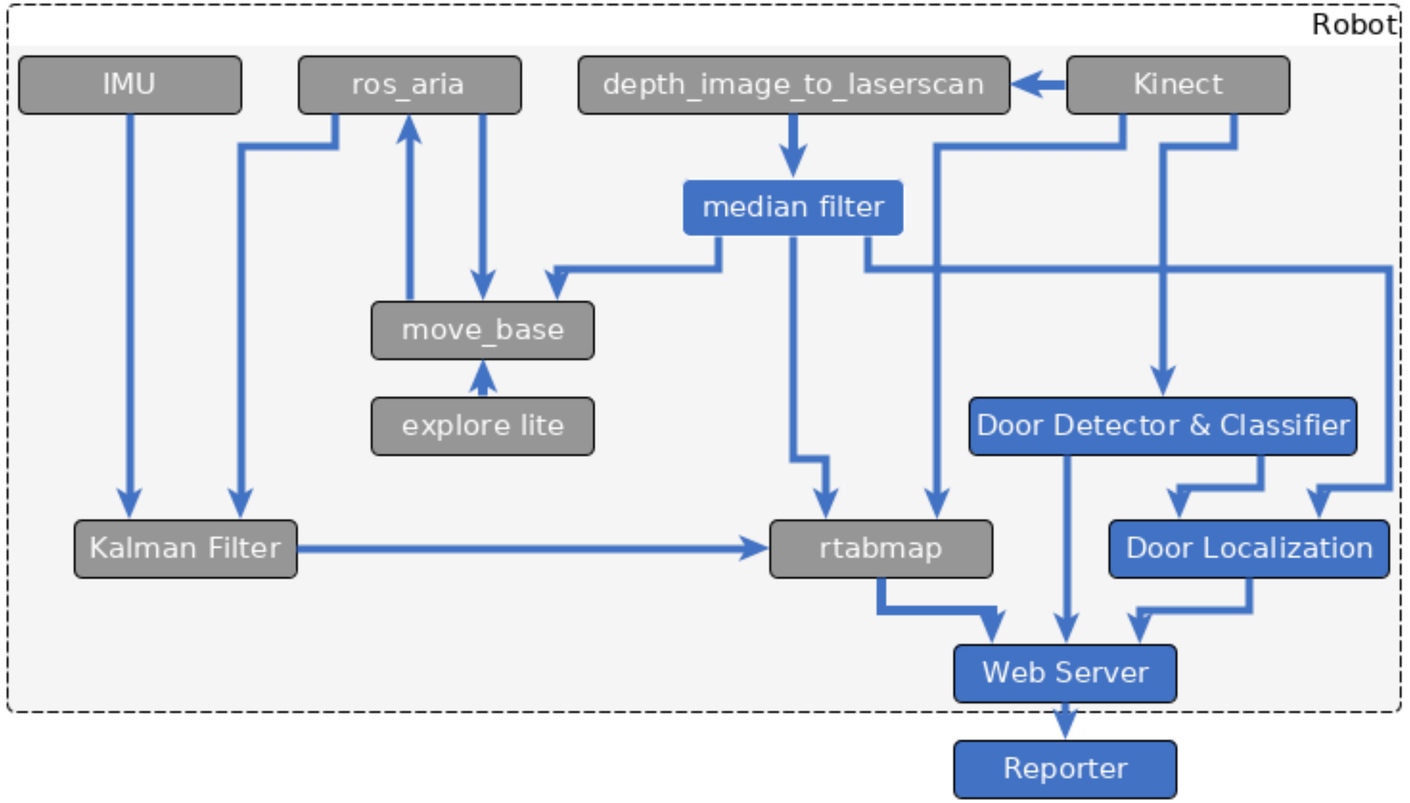
Before

After

Architecture

Modules

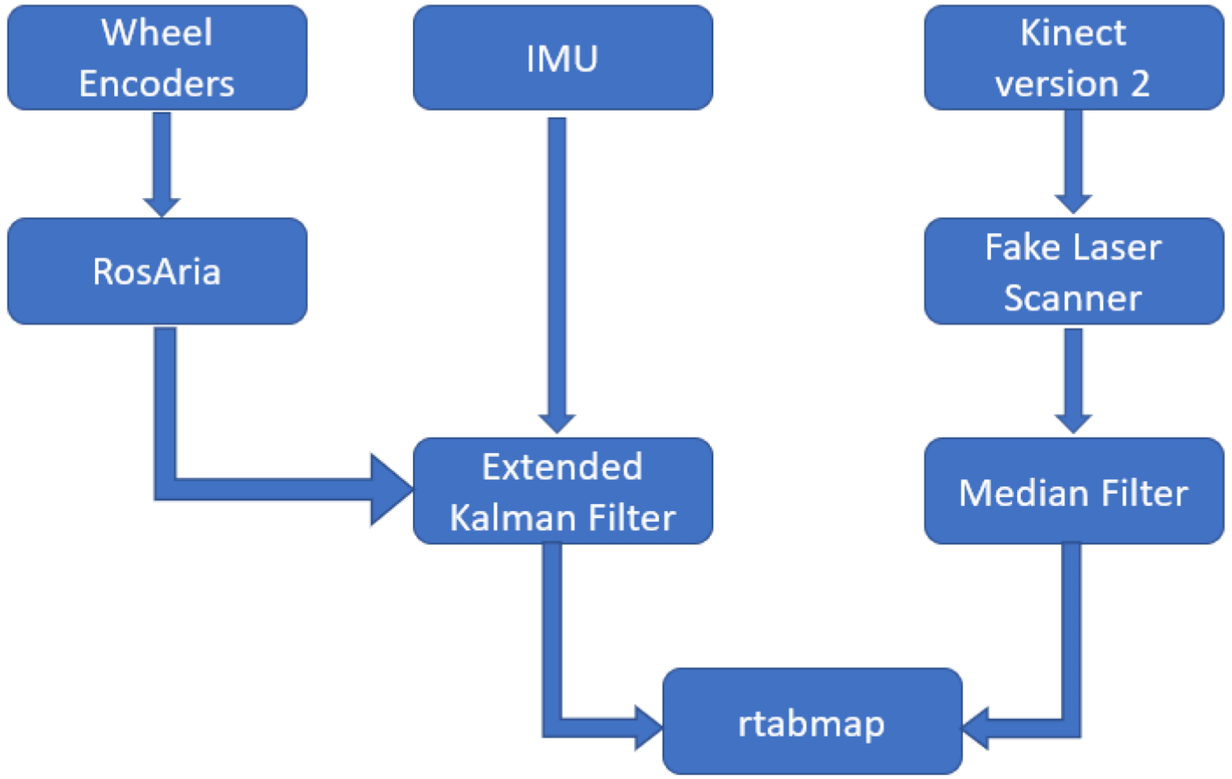
- depth_image_to_laserscan
- move_base
- rtabmap
- explore_lite
- Kinect2_bridge
- ros_aria
- robot_localization



Navigation

SLAM

- Robot localizes itself
- Generate a map
- IMU to accurately estimate rotations
- ‘Loop Closure’ to correct drifts

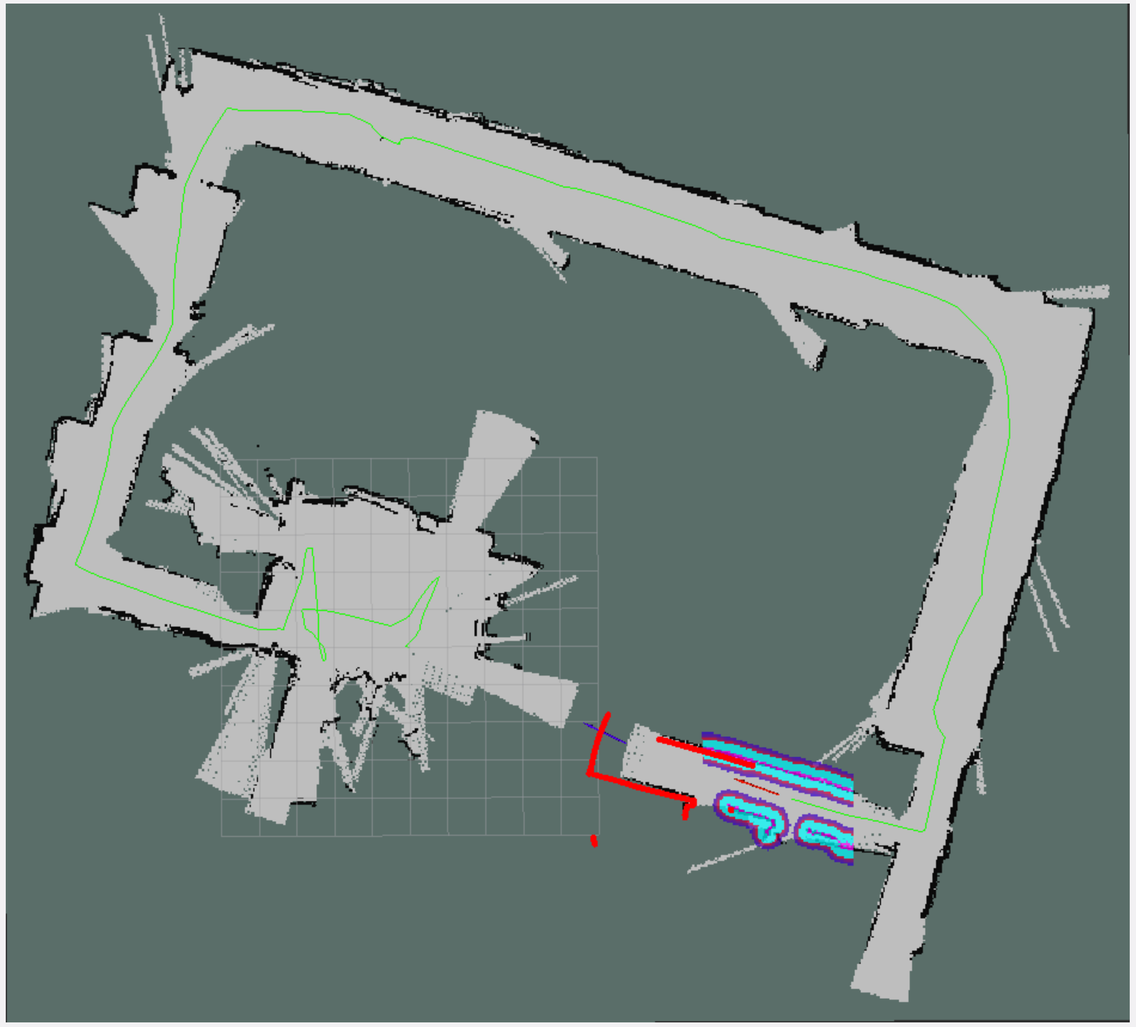


SLAM

Impact of IMU and Kalman Filter

Laser-scan + Wheel Odometry

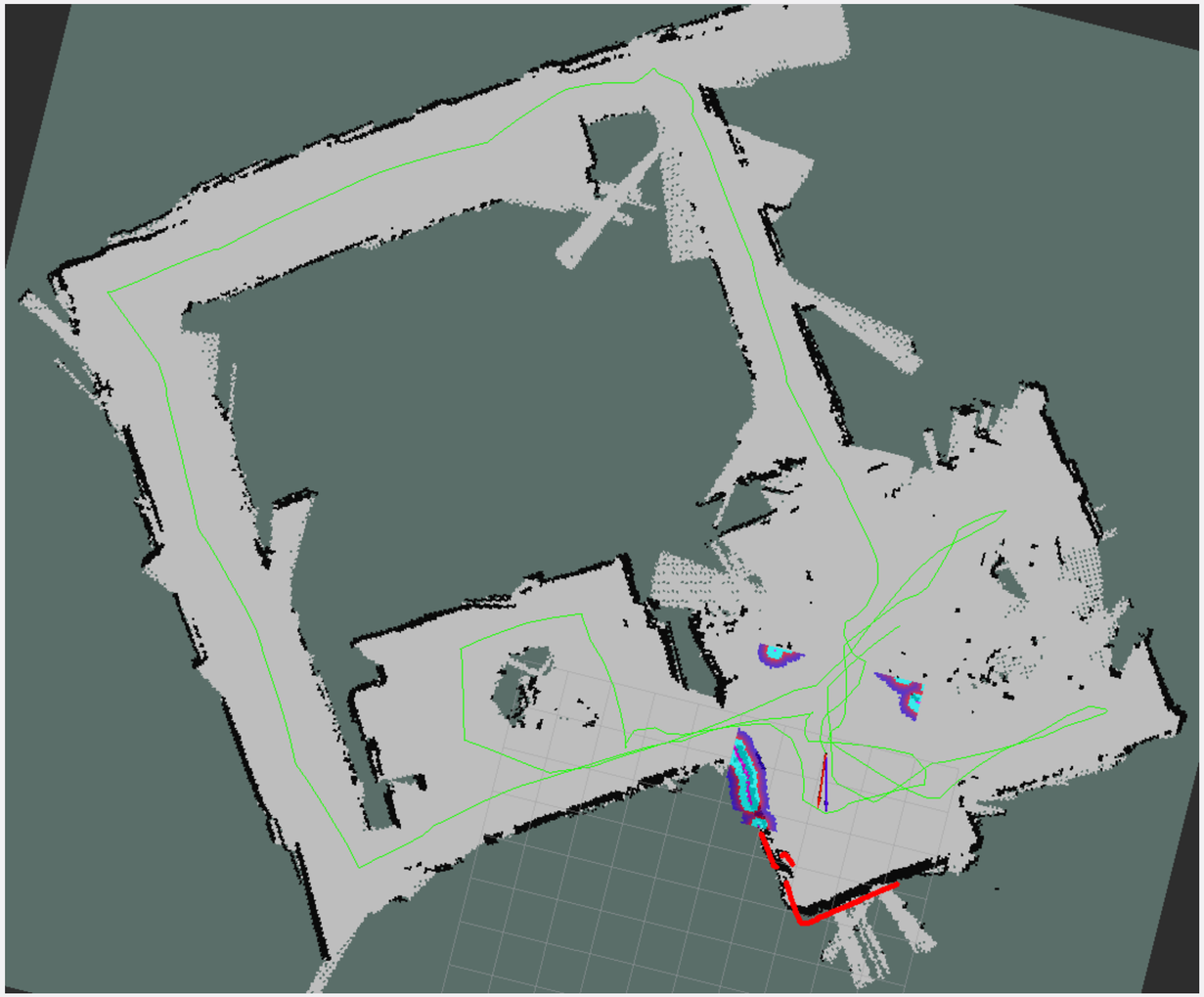
Laser-scan + Wheel Odometry + IMU



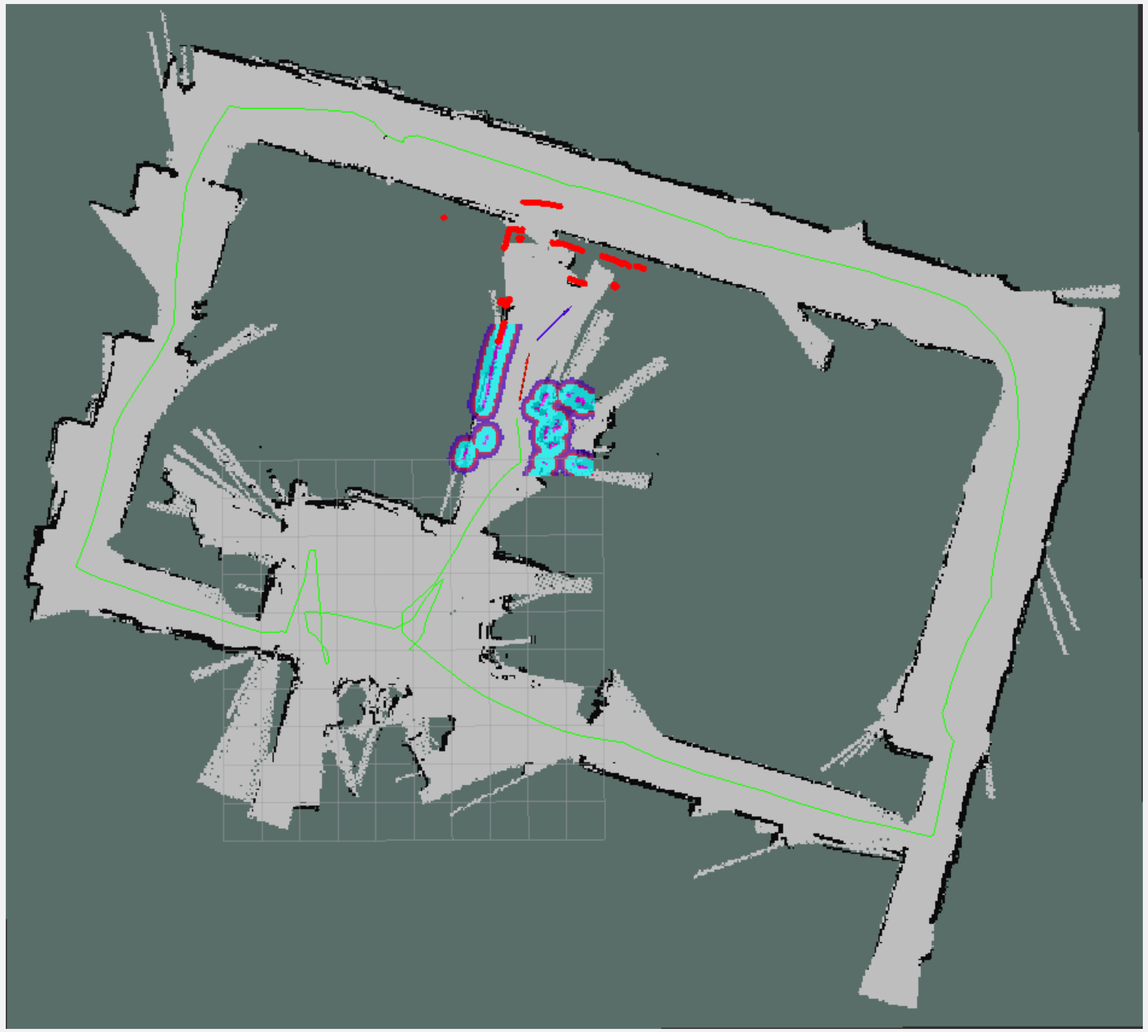
SLAM

Loop Closure

Wheel Odometry + Laser-scan

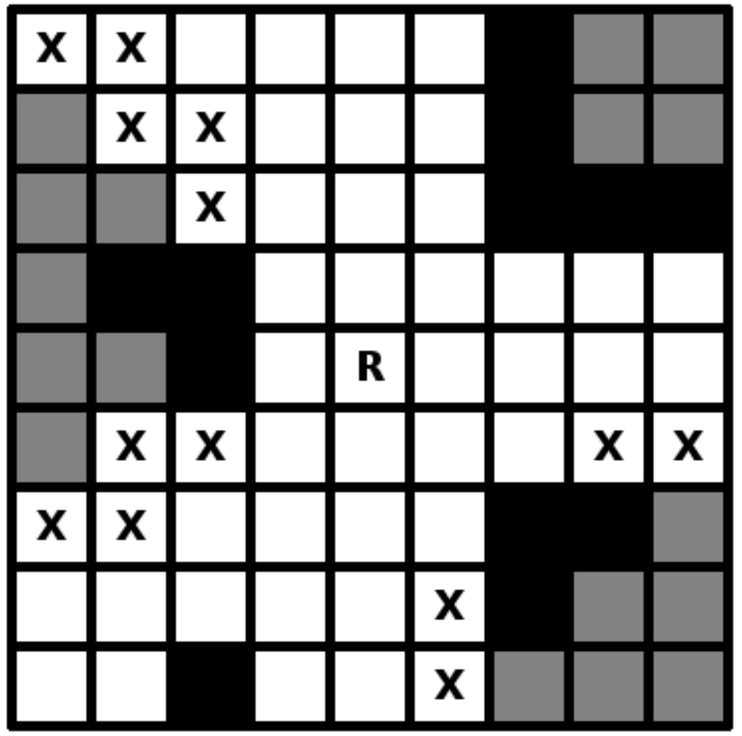


Laser-scan + Wheel Odometry + IMU



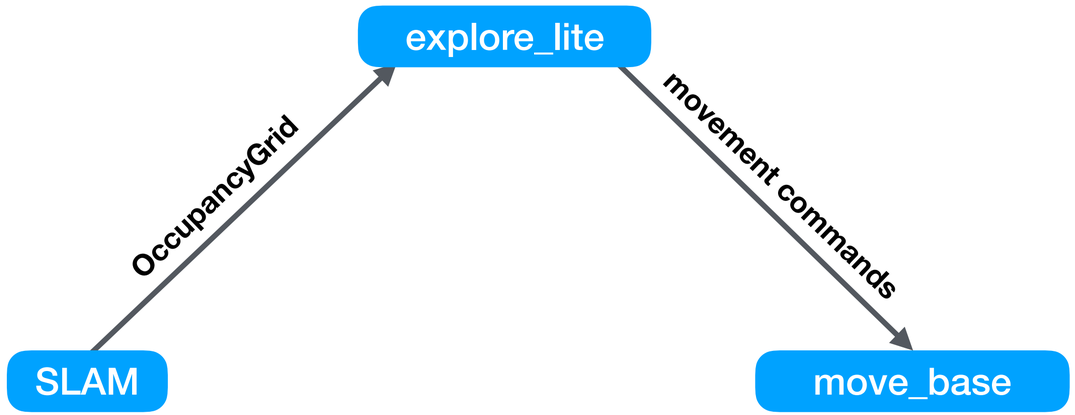
Exploration

- Frontier-based
  - R: robot
  - X: frontier
  - Black cells: occupied
  - White cells: open
  - Gray cells: unknown



Implementation

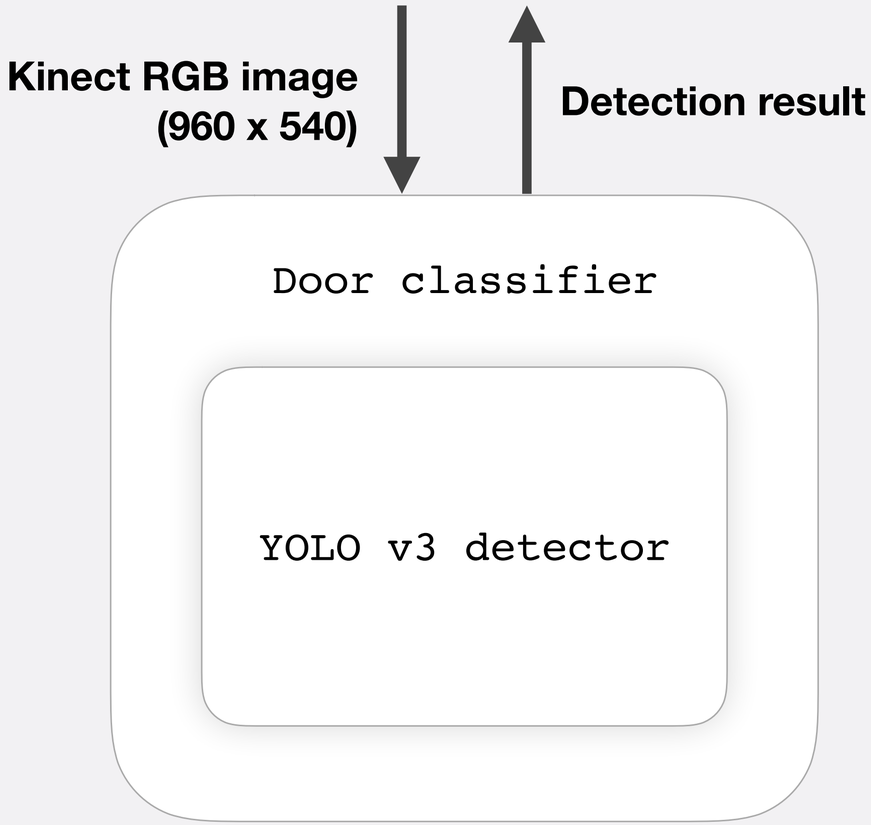
- Issues:
  - Bumping into close obstacles
  - Noise in doorway
  - Doorway too narrow after map drift



Door Classifier

Door Classifier

- ROS-node with pre-trained detection model
- Model: trained YOLO version 3
- Localizes and classifies doors in Kinect images
- Subscribes to Kinect image topics
- Publishes bounding boxes and classification results as coordinates, confidence scale and class label



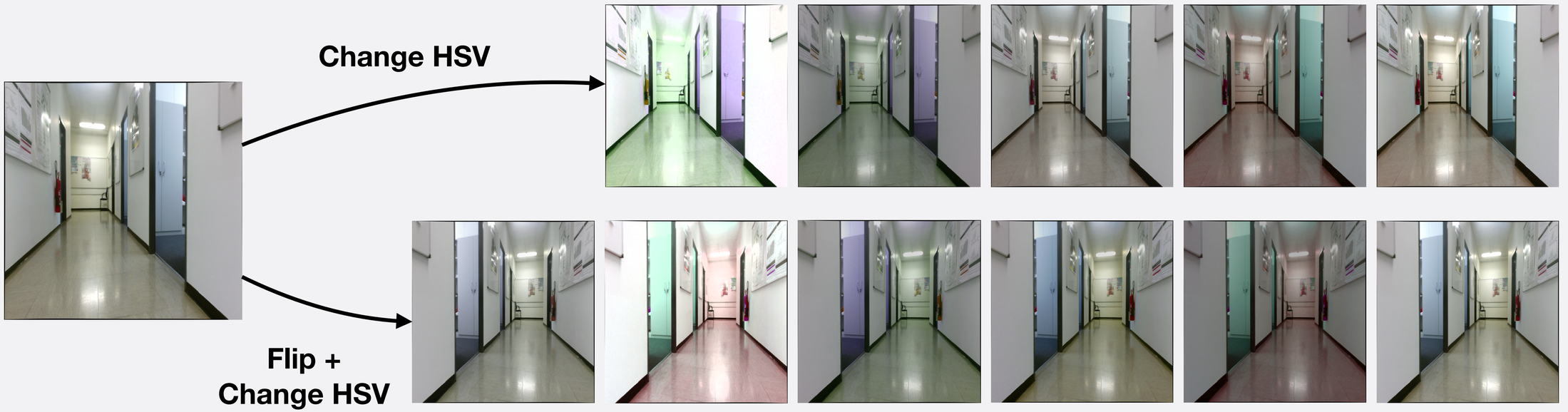
Data capturing

- 376 images captured with robot and attached Kinect v2
- Images captured all inside building R
- Image labelled with LabelImg tool
- Bounding boxes exported in YOLO format:

```
0 0.703125 0.353704 0.058333 0.705556
1 0.386719 0.3625 0.035937 0.510185
1 0.55026 0.448148 0.030729 0.118519
```

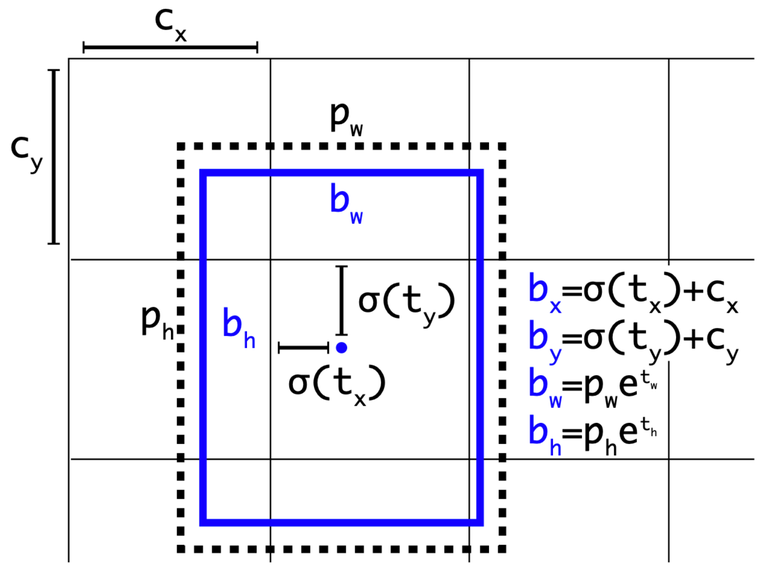
Data augmentation

- 376 - (no bounding box) * 11 = 4212 images



YOLO v3

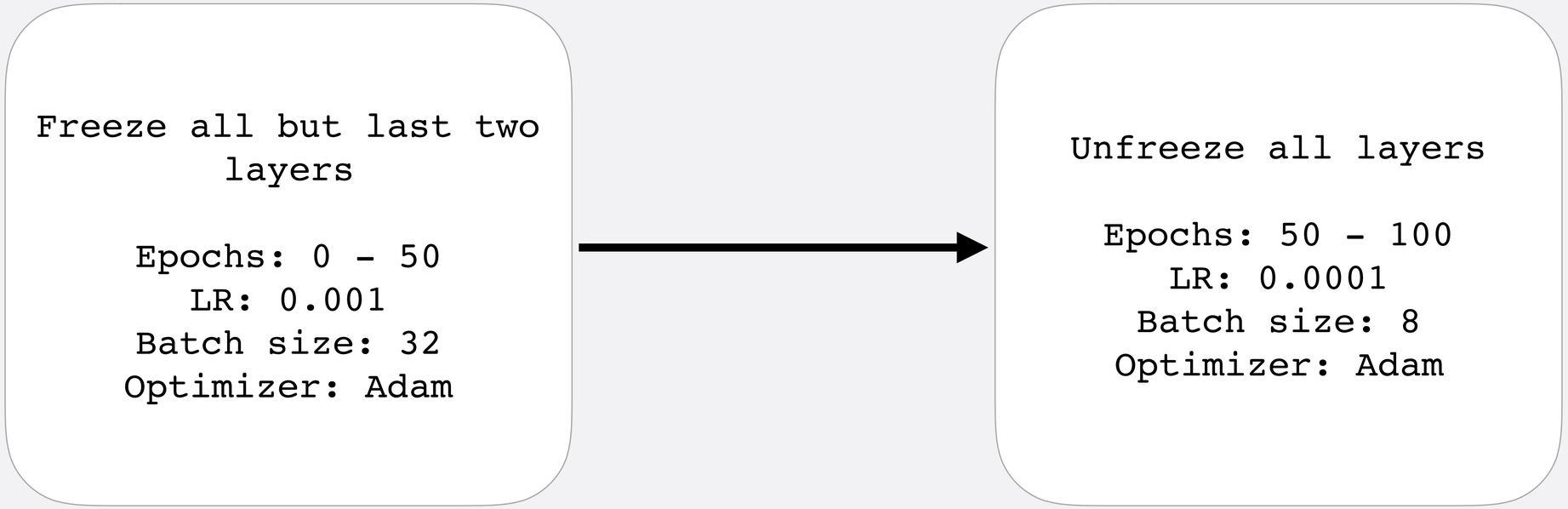
- 13 x 13 feature map
- Cell with centroid of bounding box is “responsible” for detection → Other grid cells which detect same bounding box are excluded
- Custom DarkNet-53 CNN (53 conv layers)
- Detection with *N=9* anchor boxes → per grid 9 bounding box predictions
- Maximum suppression + IoU threshold results in filtered result



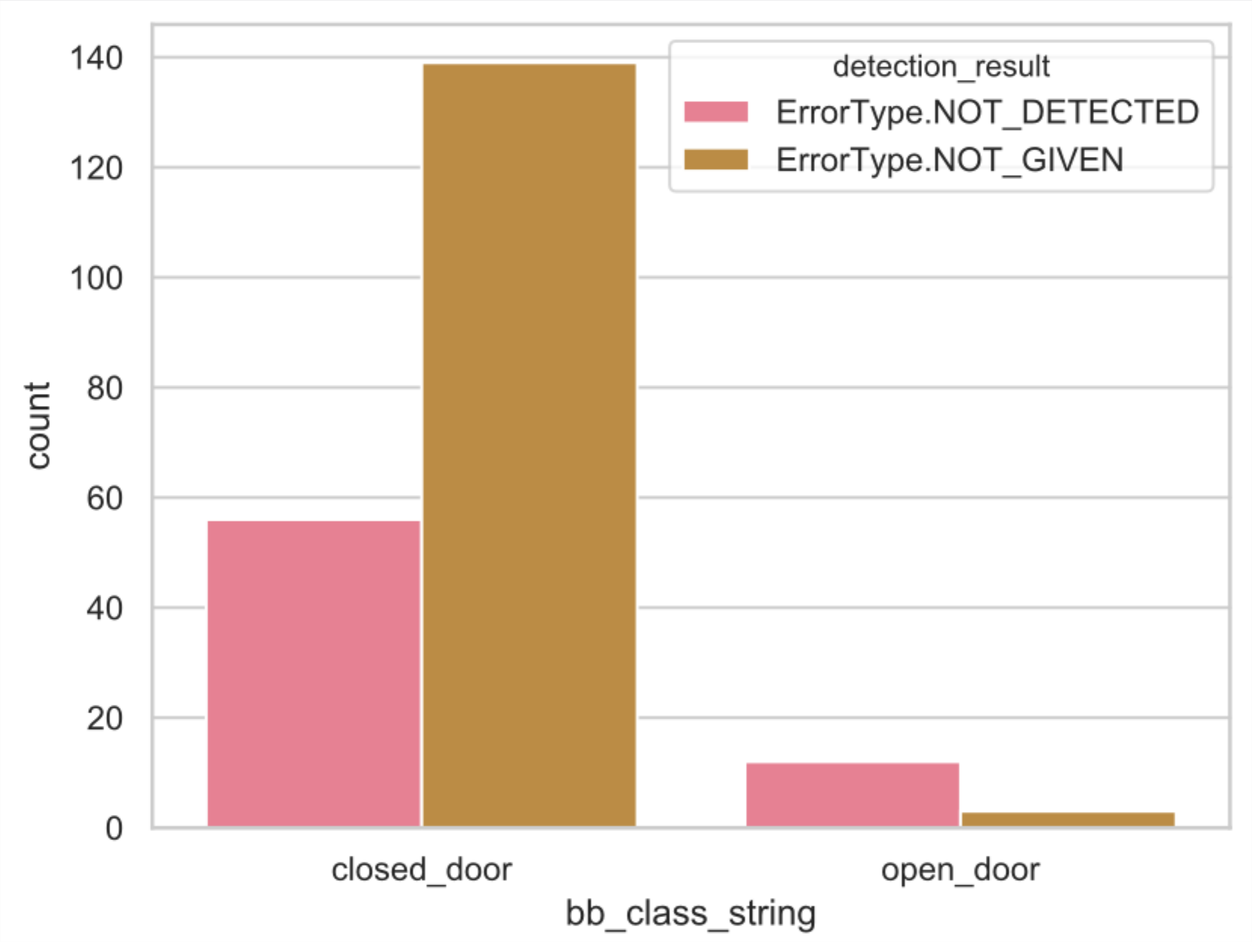
Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement.

Training

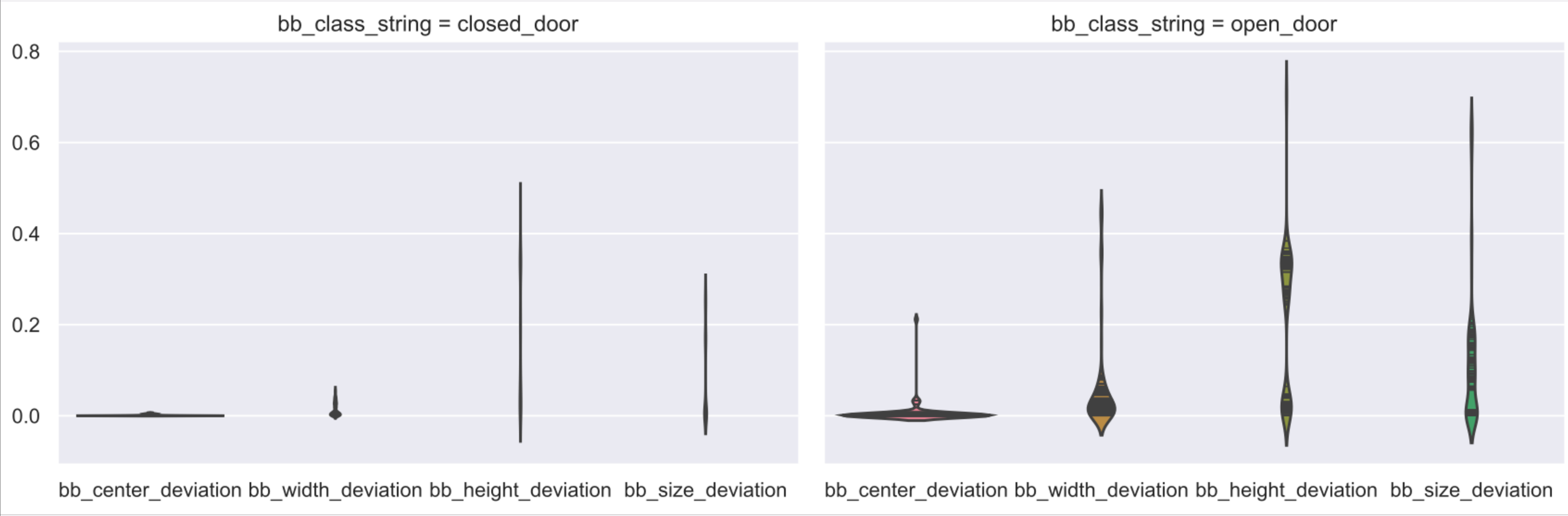
- Input shape: 416 x 416 (squeezed image)



Experimental Results



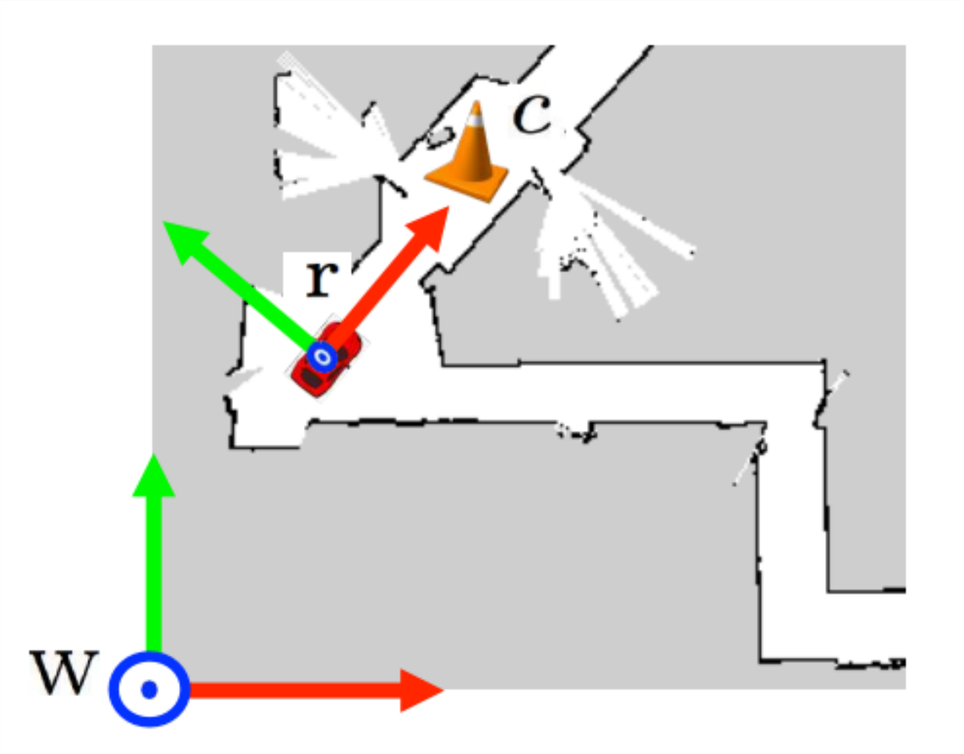
Experimental Results



Door Position

Rigid Body Transformation

- Transform points from one coordinate to another



Relative Door Position

- Using Camera Matrix P: $P = \begin{bmatrix} f_x & & c_x \\ & f_y & c_y \\ & & 1 \end{bmatrix}$

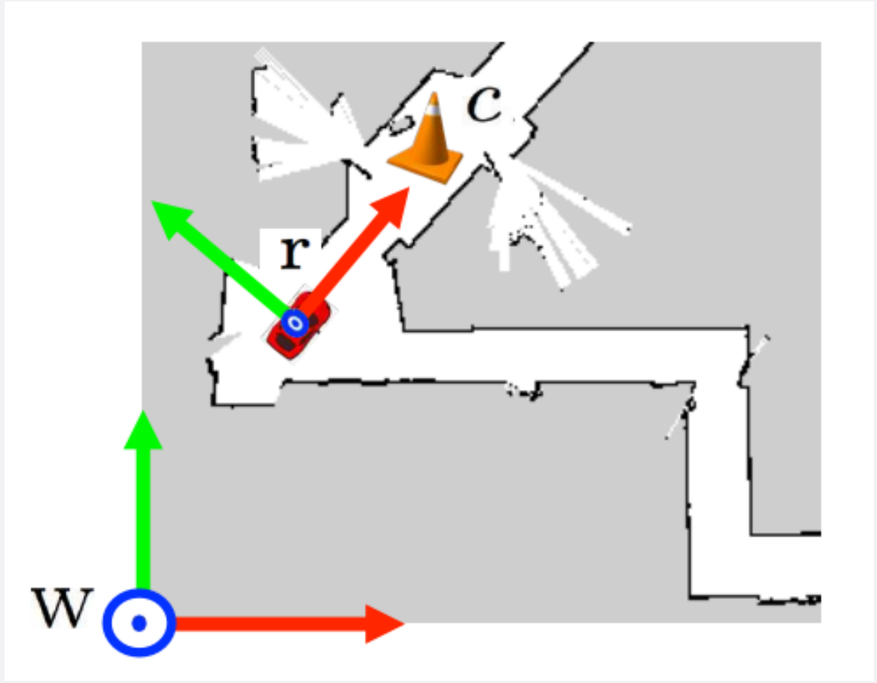
- With depth value, Project (3D to 2D):

$$
\begin{aligned}
u &= \frac{f_x x + c_x z}{z} \\
v &= \frac{f_y y + c_y z}{z} \\
d &= z
\end{aligned}
$$

Absolute Door Position

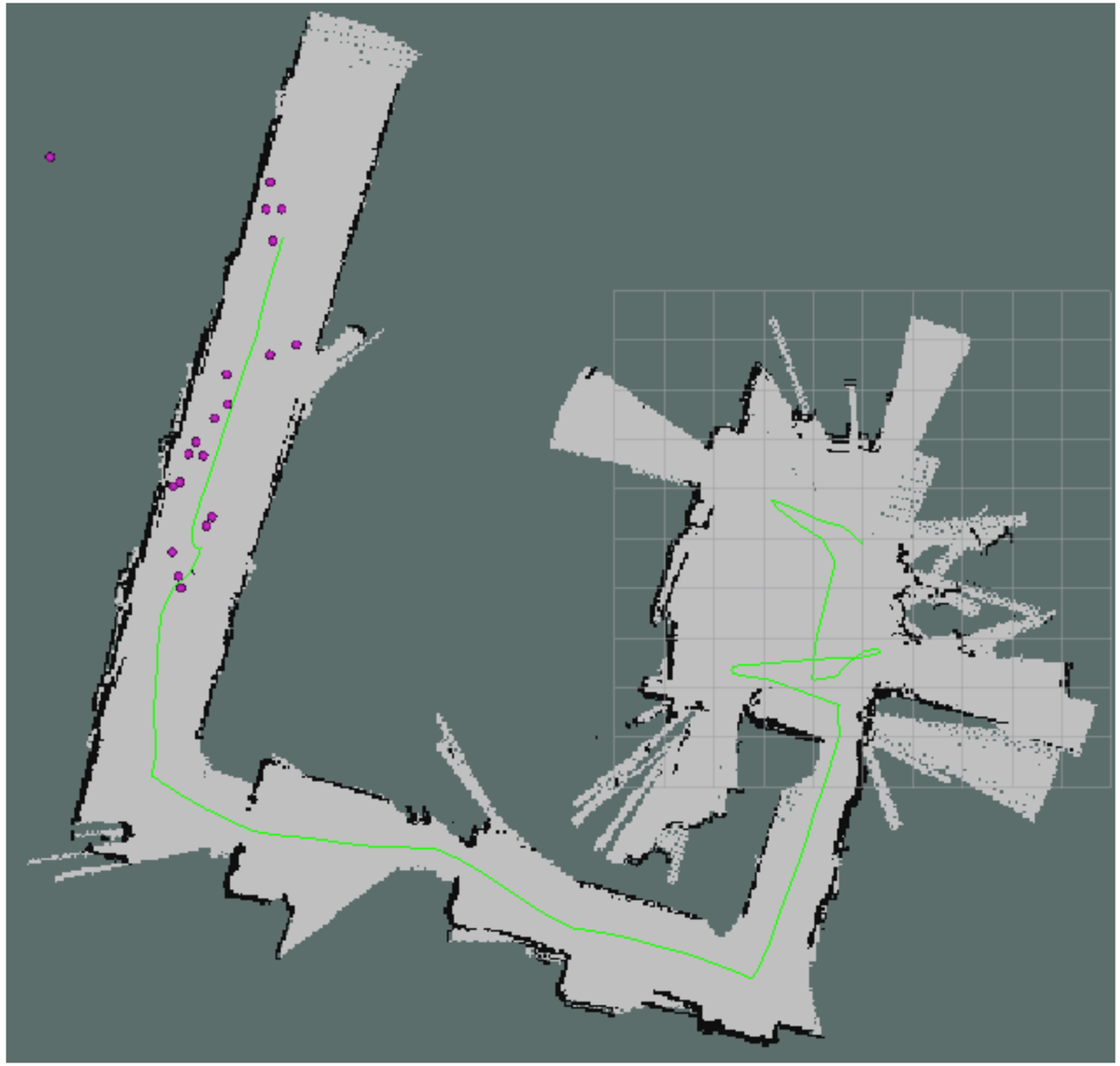
- Given point c in frame r;
  - What is the position of the point in frame W?

- $$p_c^w = R_r^w p_c^r + p_r^w$$



Calculated Door Positions

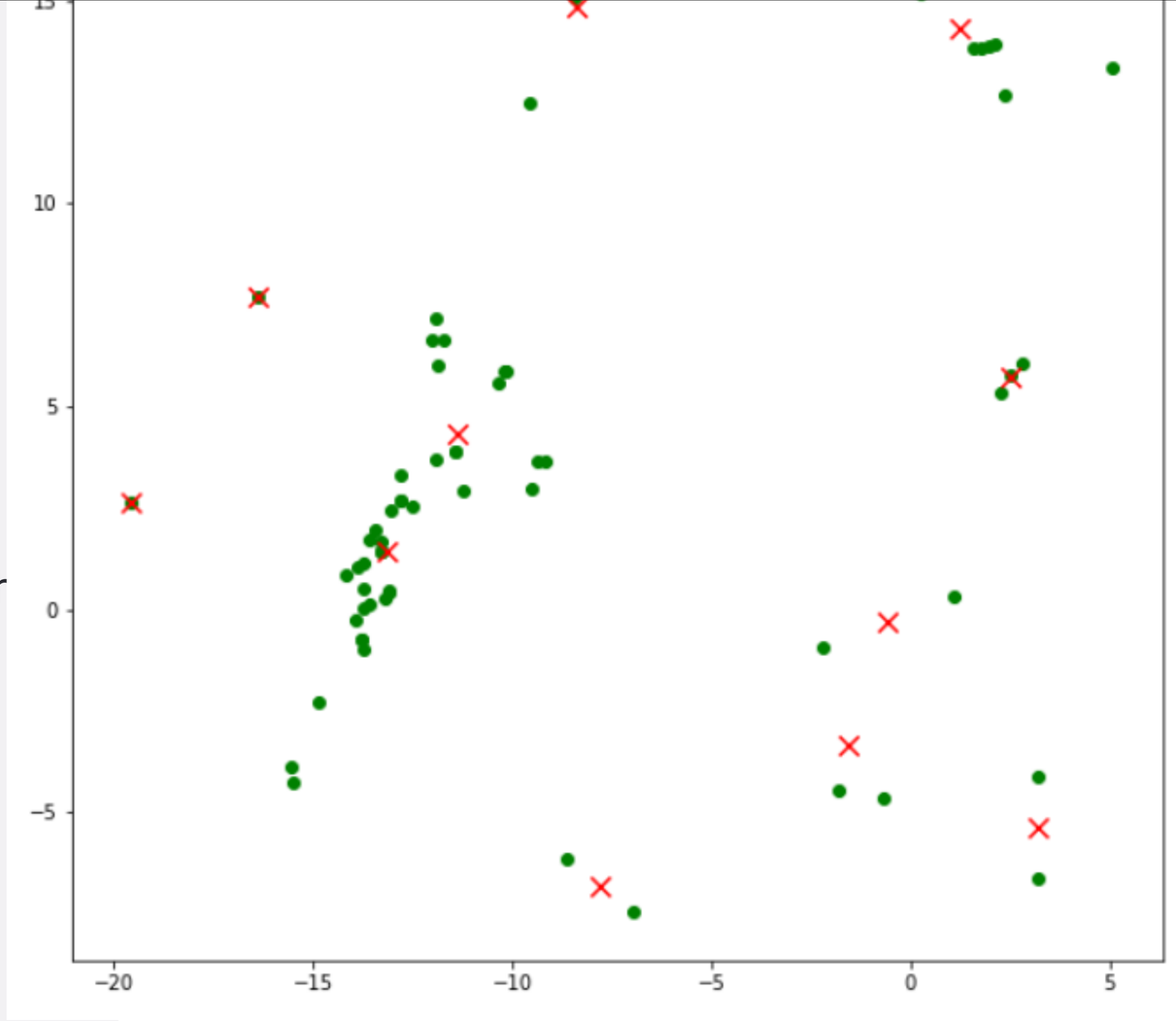
- Calculated door positions
- Depicted on world map



Subtitle

Clustering (Mean shift)

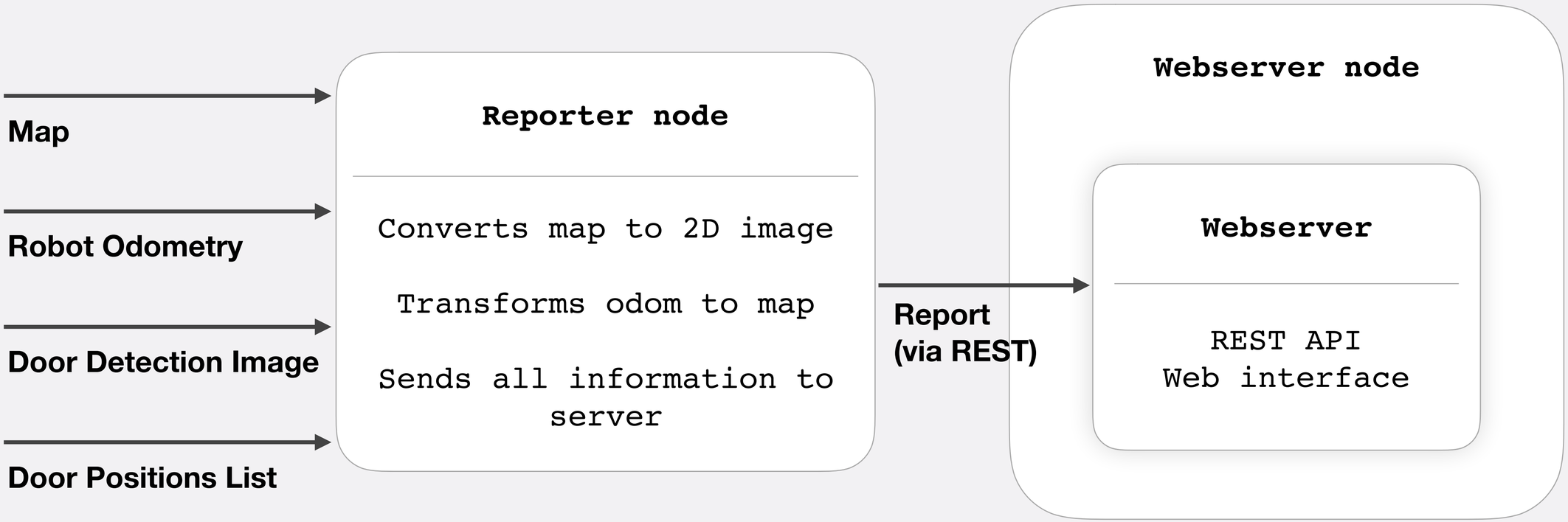
- Centroid-based algorithm
- Updates candidates for centroids as th mean in a given region
- Representation of how mean shift clustering works:



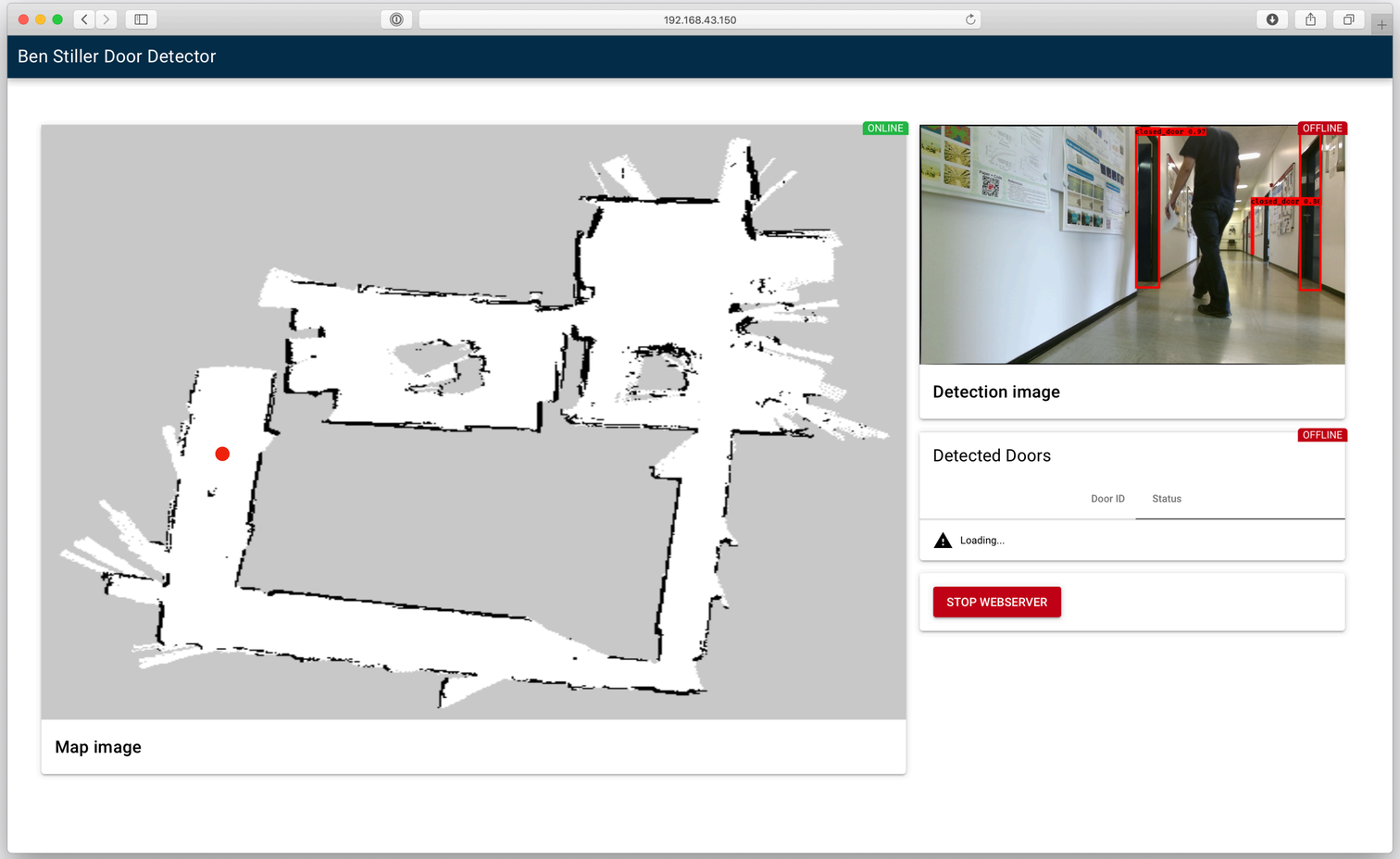
Reporter + Web interface

Reporter + Web interface

- 2 ROS-nodes: Reporter and Webserver



Web Interface



Live Demonstration

Conclusion & Discussion

Conclusion & Discussion

- Hypothesis was proved

  We can use computer vision techniques in both navigation and semantic understanding of the environment to design an autonomous guard.

- Component "night vision” as next step

  Convert the depth point cloud into black and white images and use these for loop closure and training the door detector.

- Door clustering algorithm needs to be tested more thoroughly

  such that the detected doors are not placed on the reporter map as duplicates.

References

- Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767.