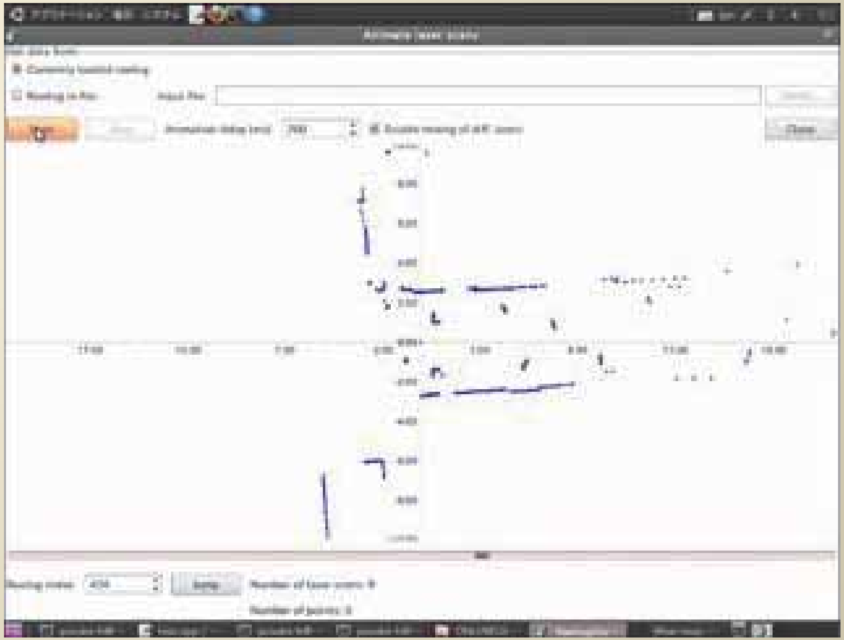
Real-time Scan Matching Using L0-Norm Minimization Under Dynamic Crowded Environment

Y. Hieida, T. Suenaga, K. Takemura,
J. Takamatsu, T. Ogasawara
Nara Institute of Science and Technology, Japan

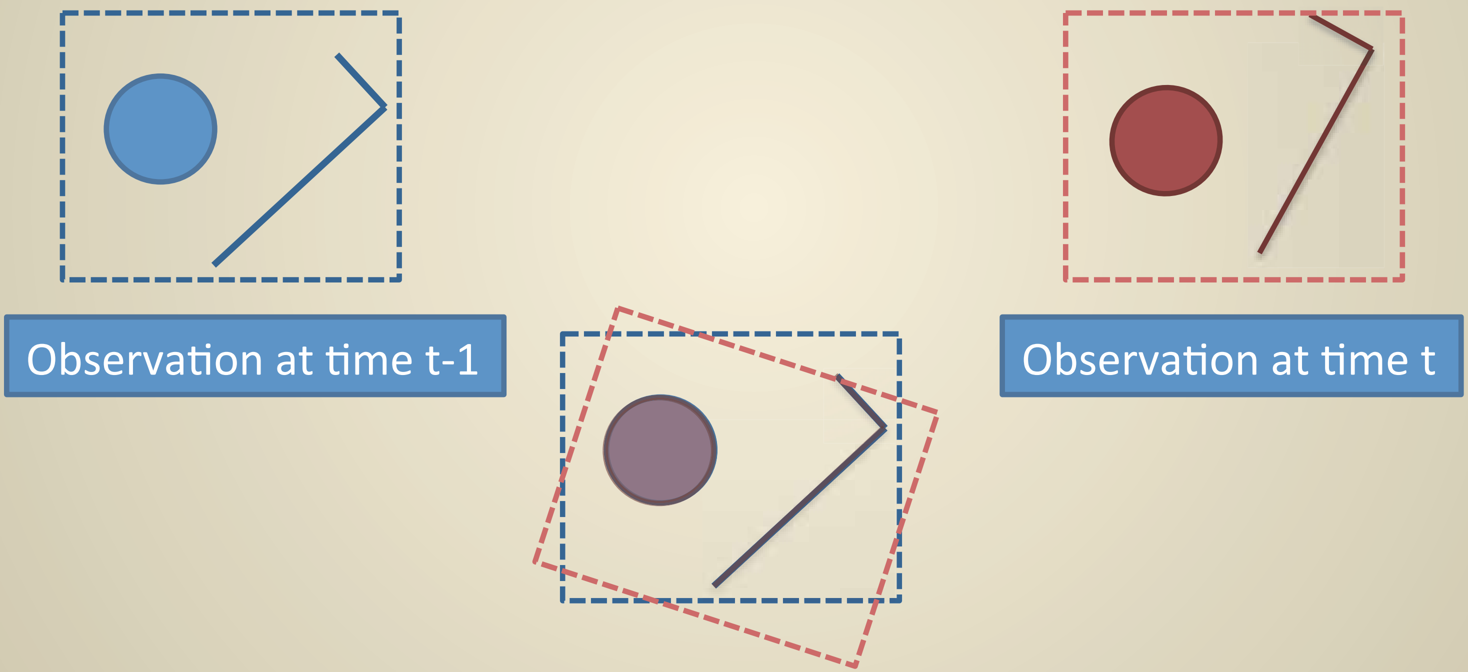
Our Goal

- Realize autonomous mobile robots under crowded scene



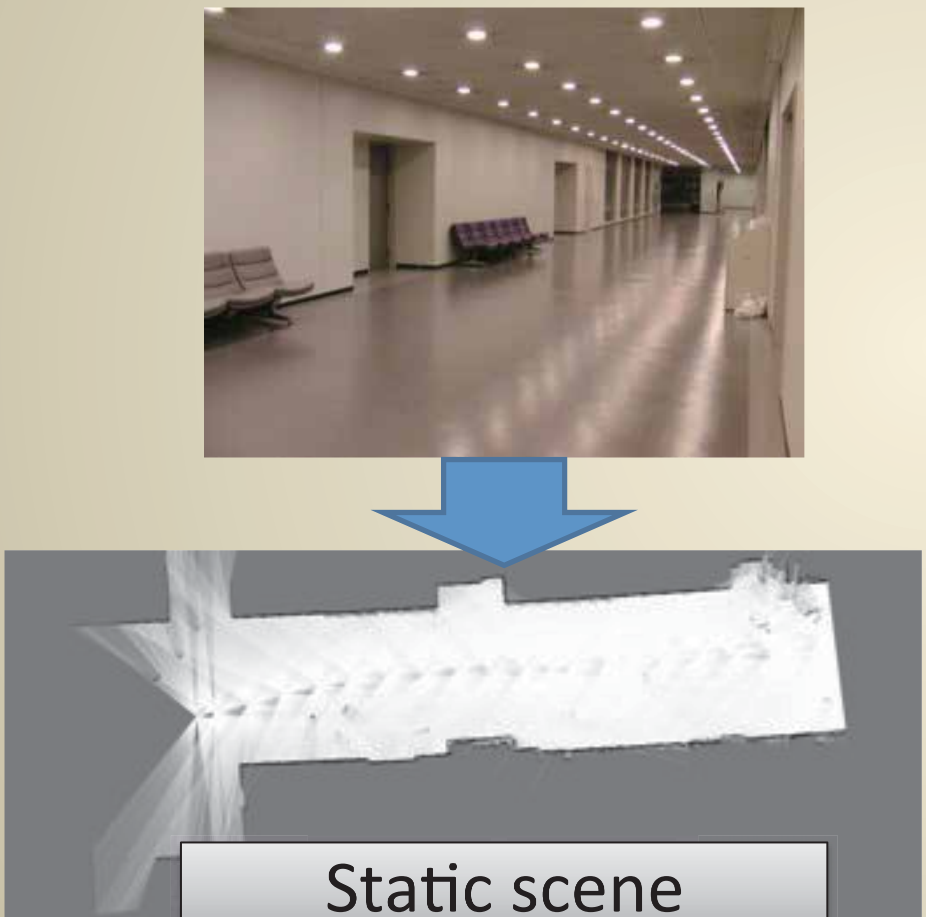
Localization and mapping are important

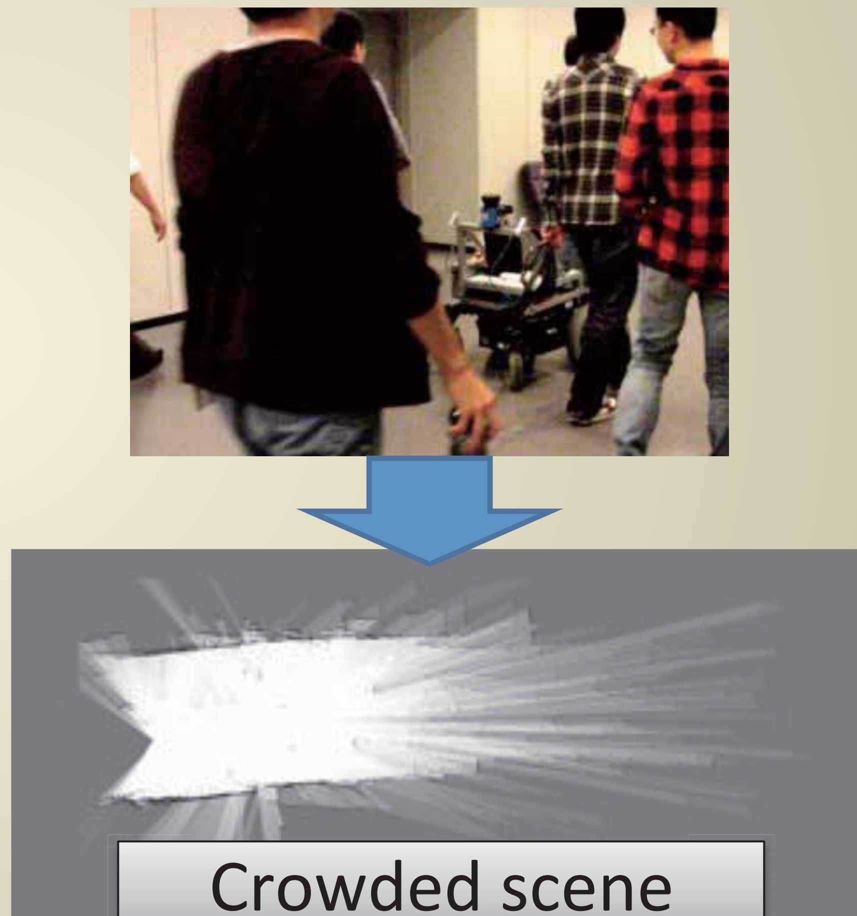
ICP-based Scan Matching



Alignment gives us estimation of displacement

Issues





Related Works

- Use robust estimator (e.g. M-Estimator)

- Remove moving objects
  - Use landmarks [Wolf et. al, 2005]
  - Use tracking [Hahnel et. al, 2002] [Wang et. al, 2003]

- Solve moving object detection and localization simultaneously
  - EM-algorithm [Hahnel et. al, 2003]
  - Use integrated probabilistic model [Ven et. al, 2010]

Proposal

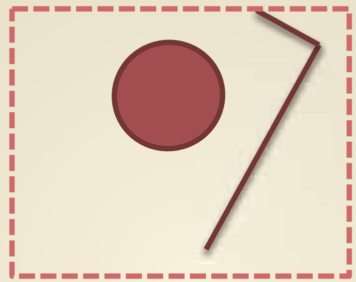
- Use L0-norm, a kind of robust estimator

- Accelerate by Locality Sensitive Hashing
  →Keep real-time (about 5 fps)

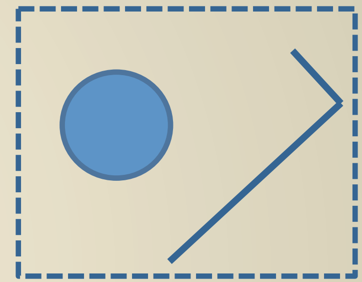
PROPOSAL

Flow of Algorithm

1. Observe environment by LIDAR

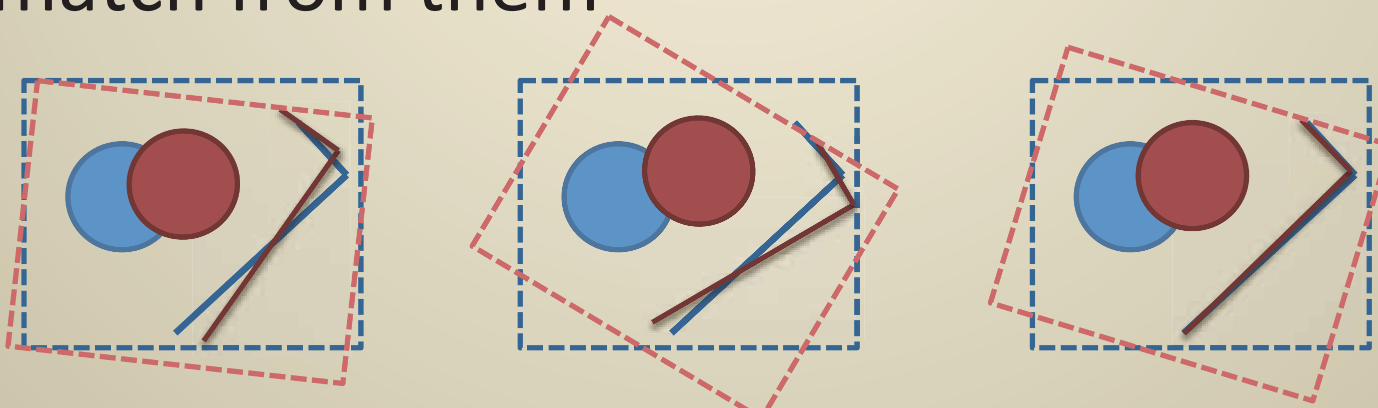


Observation at time t



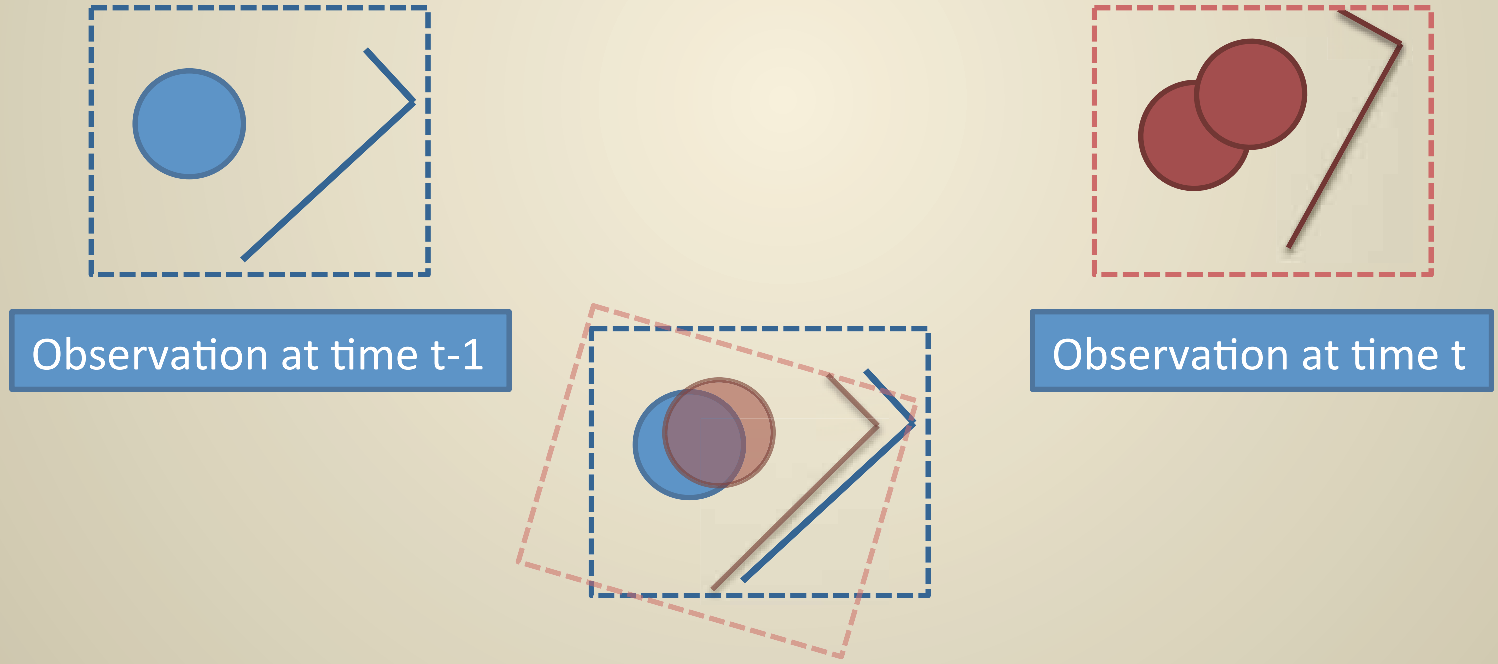
Observation at time t -1

2. Generate candidates and select the best match from them



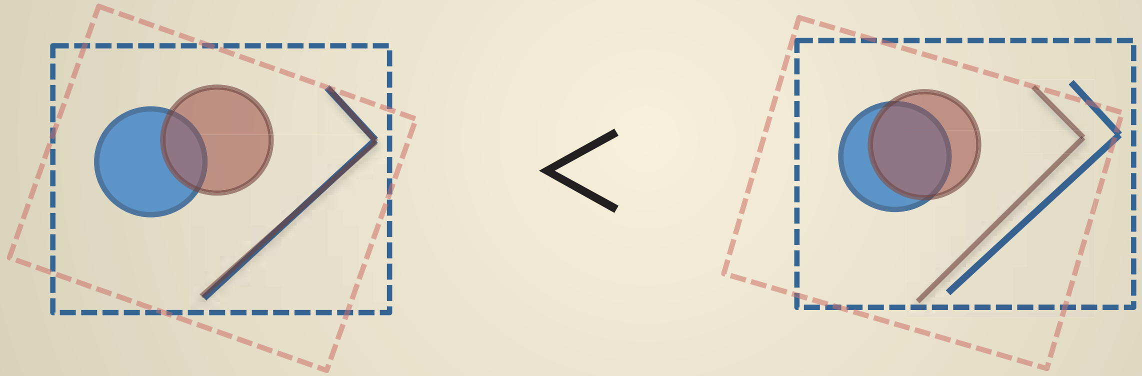
Review of ICP

- Minimize sum of squared distances of closest points



L0-Norm

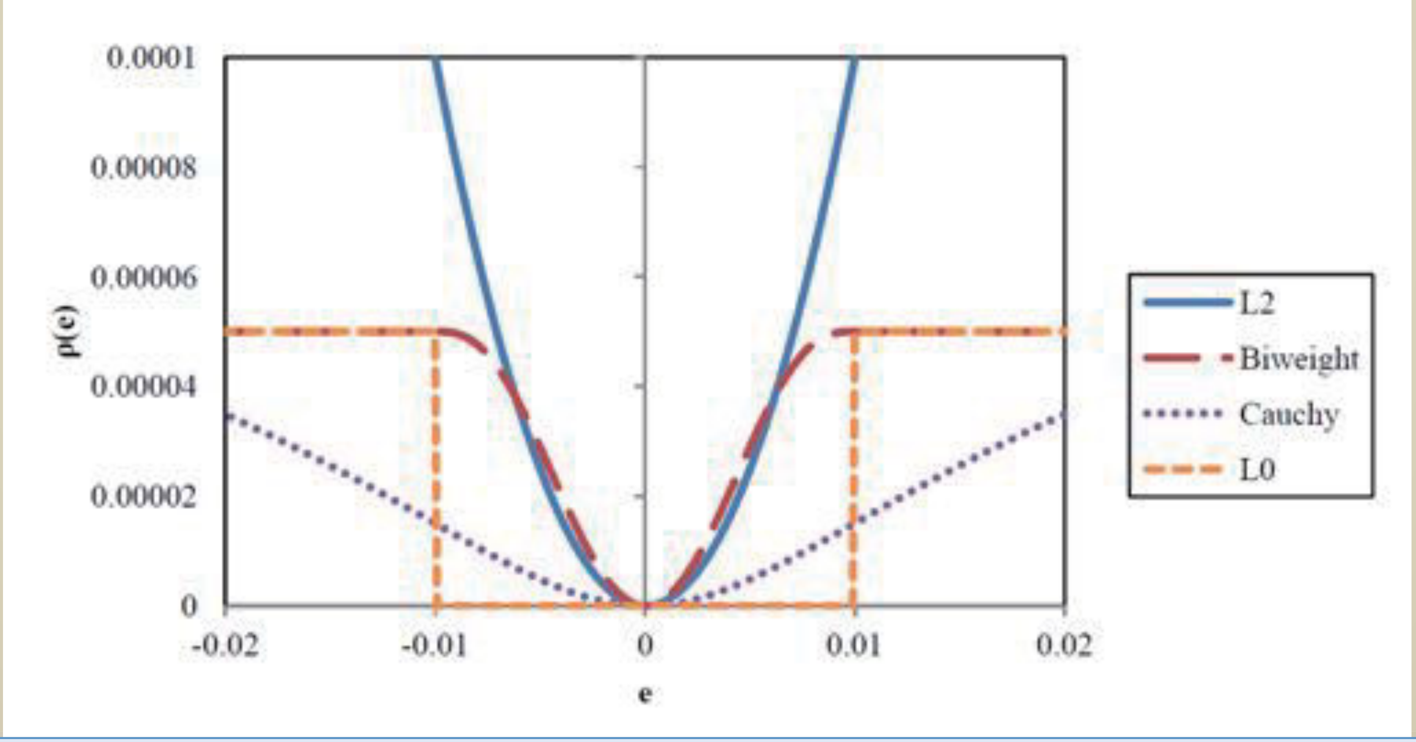
- Minimize number of not-matched points



Assumption: static objects are dominant

Issue in L0-Norm Minimization

- Not use gradient information

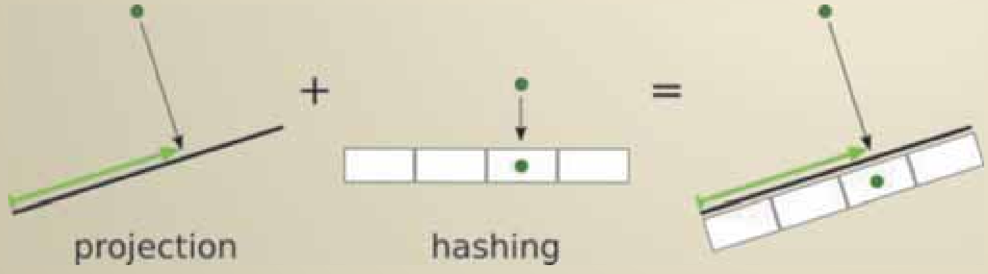


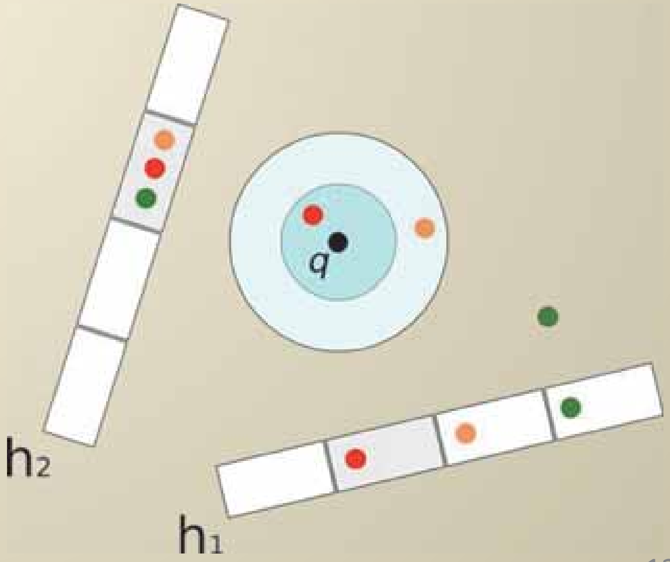
Give up gradient-based optimization and use brute-force search

Locality Sensitive Hashing (LSH)

- For approximate closest point search
- Use multiple hash functions
  - Limit search range by hash values

Hash functions





Difference in L0-Norm Scan-Matching

- Search matched points, not closest points

- Not use location and index of the matched point

To know existence is only important

Scenario in LSH (1)

- All hash values are different

→ Closest point: continue to search points with similar hash values

→ L0-norm: no matched-point exists

Scenario in LSH (2)

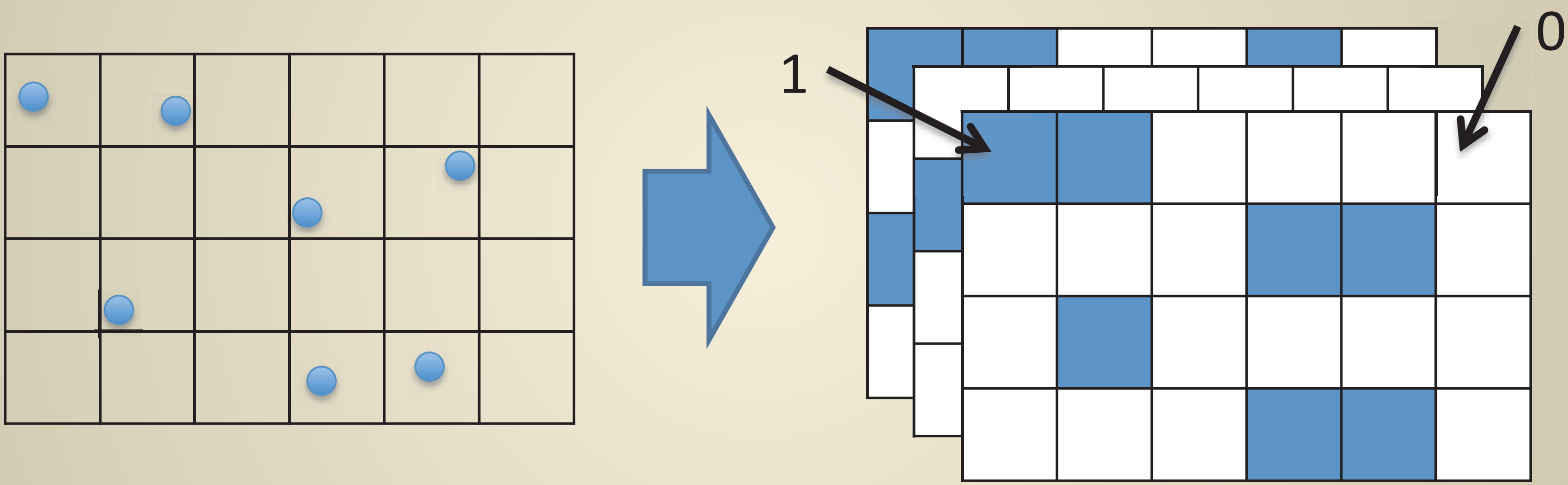
- Multiple points have the same hash values

→ Closest point: select the best one by additional calculation

→ L0-norm: multiple matched-points exist

LSH for Scan-Matching

- Use a lookup table



Advantage

- Just access memory for calculation
- Binary is sufficient for each bin

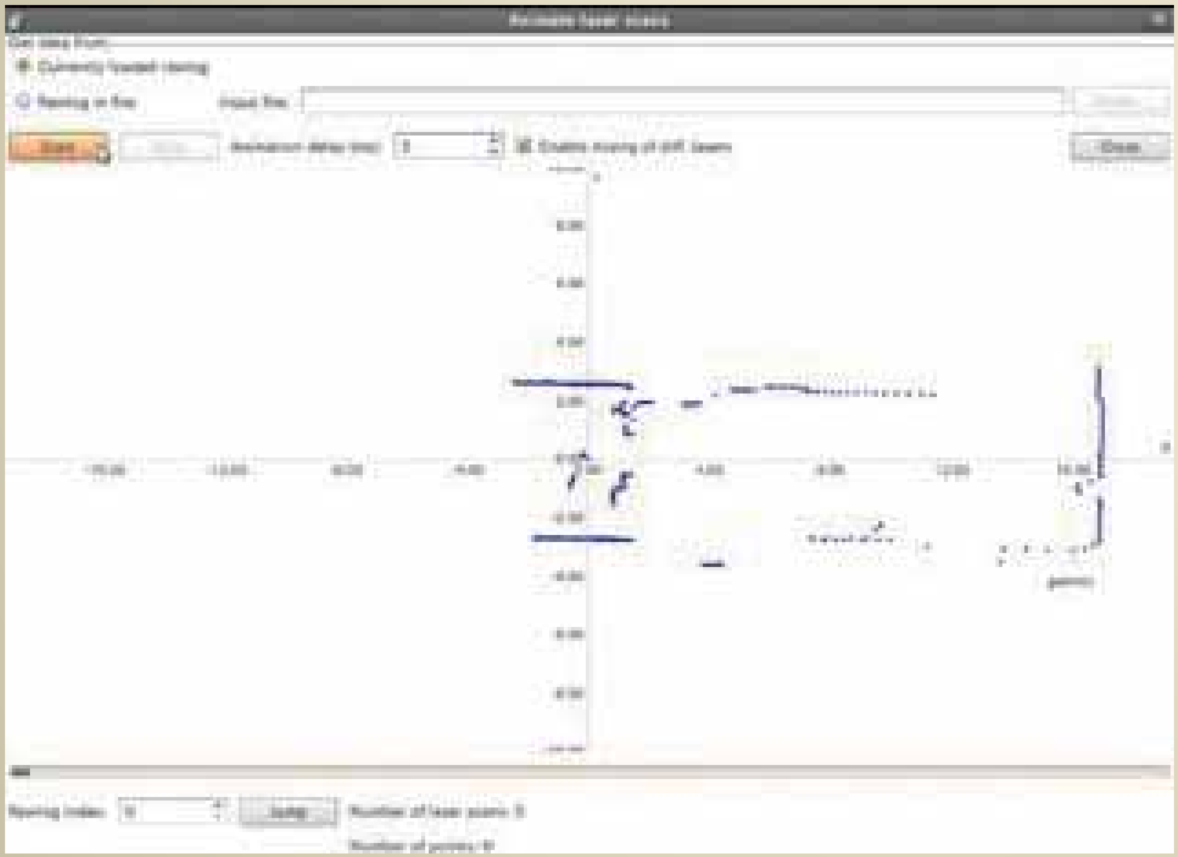
EXPERIMENT

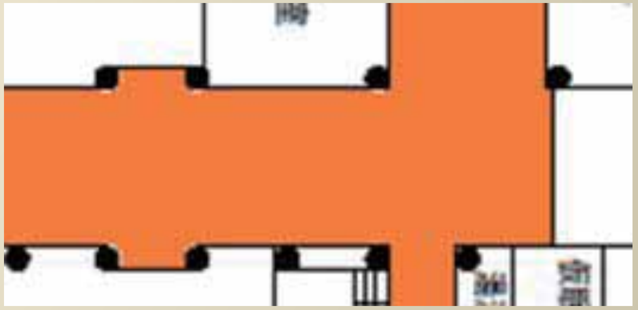
Purpose of Experiments

- Robustness to moving objects
- Accuracy in map generation

Setup of First Experiment (Robustness)

- About 25 persons move around the robot fixed

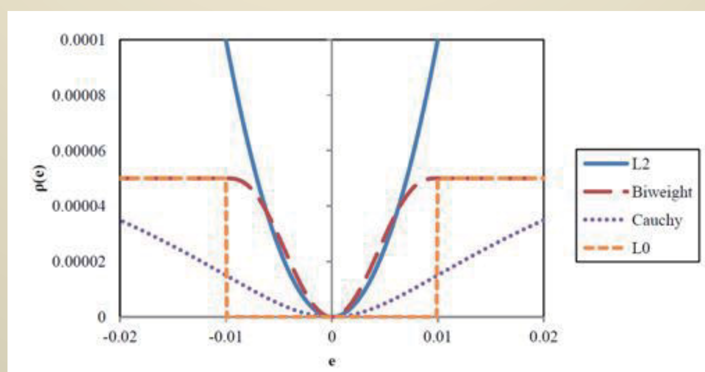




Result

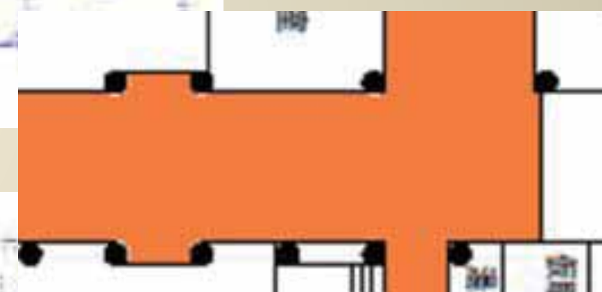
Method	RMSE (x [cm], y[cm], Θ [rad])
L2-norm	(3.37, 2.08, 0.016)
Threshold (30 percent)	(1.17, 0.033, 0)
Thresh[illegible]	[illegible]
Cauchy	(1.33, 0.013, 0)
Biweight	(0.13, 0.002, 0)
L0-norm	(0.42, 0.015, 0)

Verify plausibility of use of L0-norm

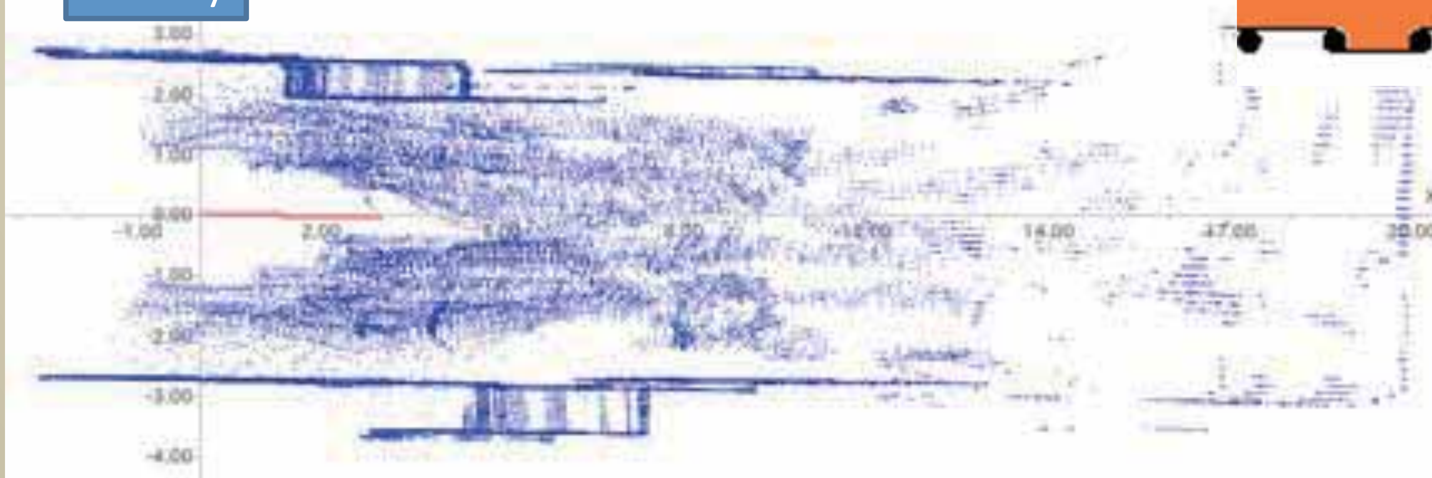


Estimated Maps

Threshold (30 percent)

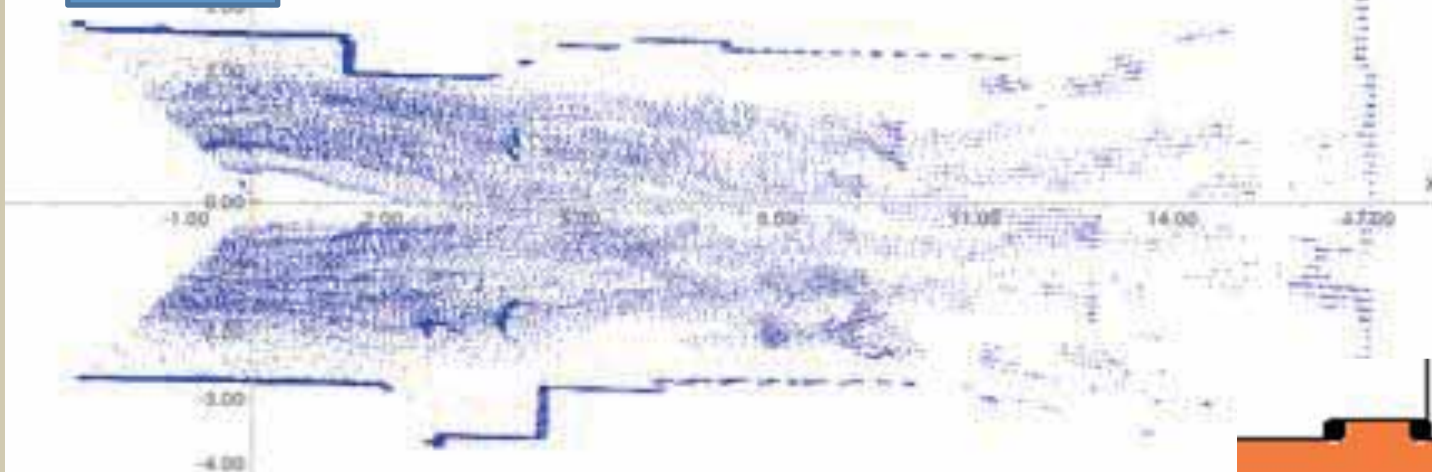


Cauchy

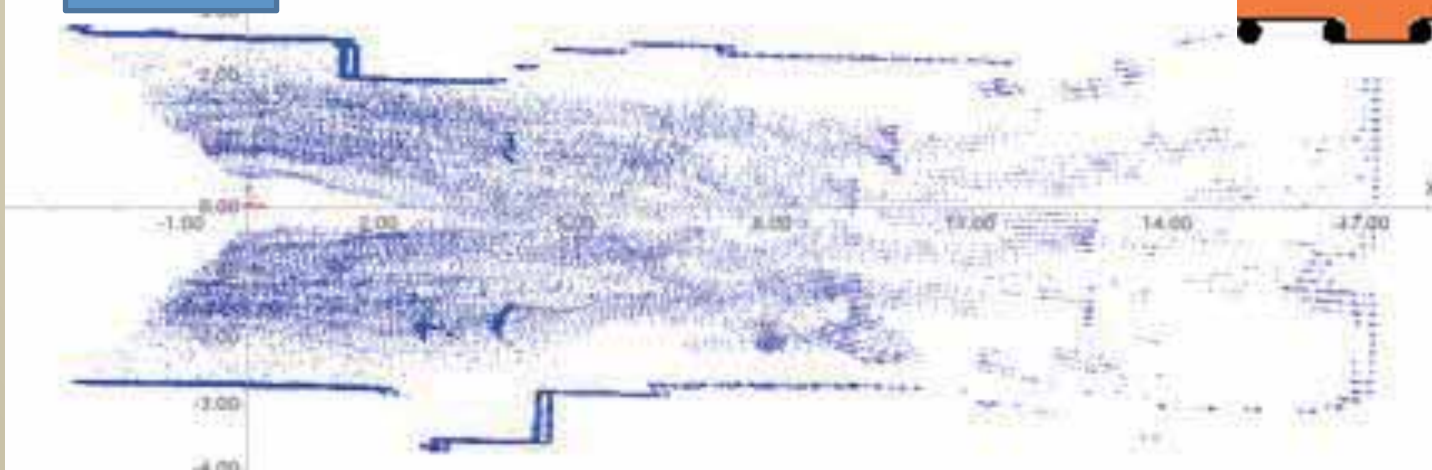


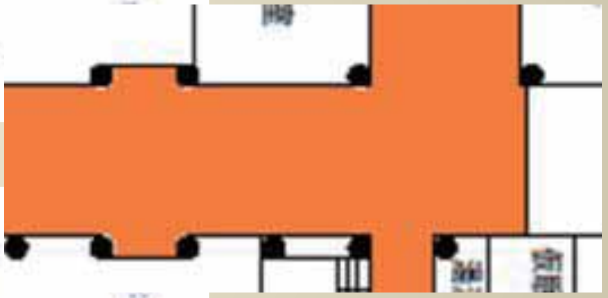
Estimated Maps

Biweight



L0-Norm





Calculation Speed

Method	# of localizations per second	RMSE
Brute-force	190	-
kd-tree	1000	-
LSH	4430	14.42

ICP, Biweight

L0-norm

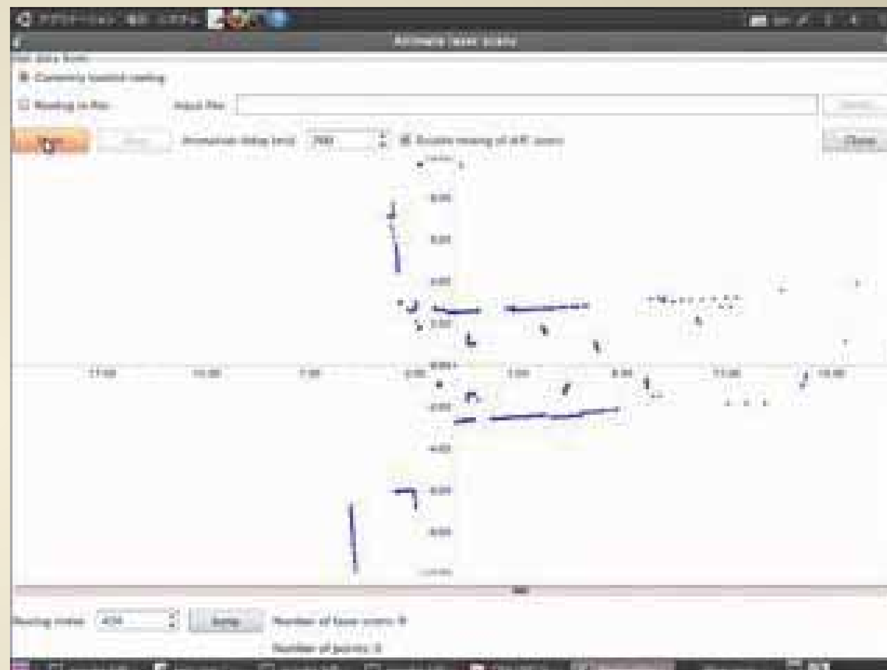
More accelerated by parallel computing

Setup in Second Experiments (Accuracy)

- About 25 persons move around the moving robot

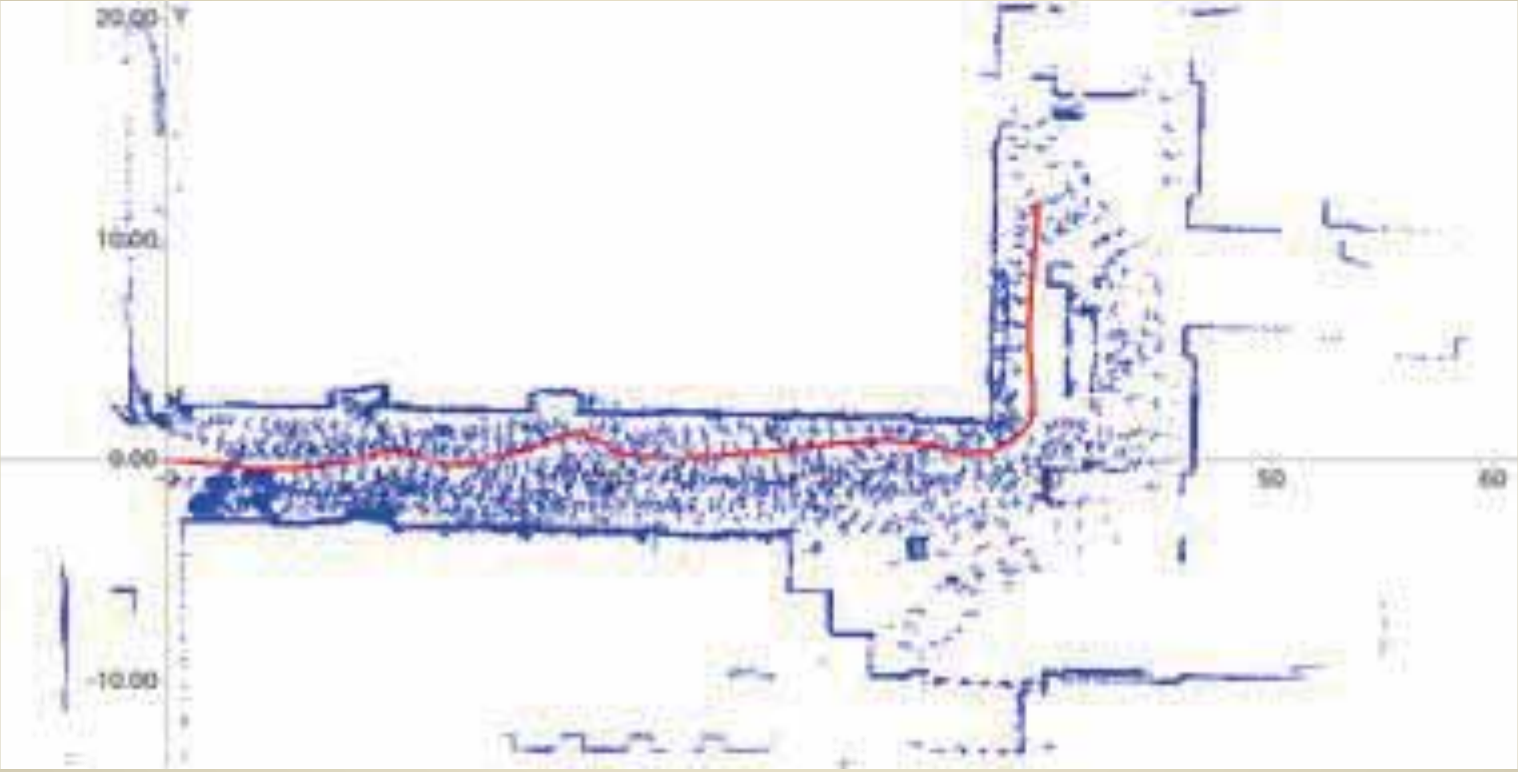
Experimental Scene

Views from the robot



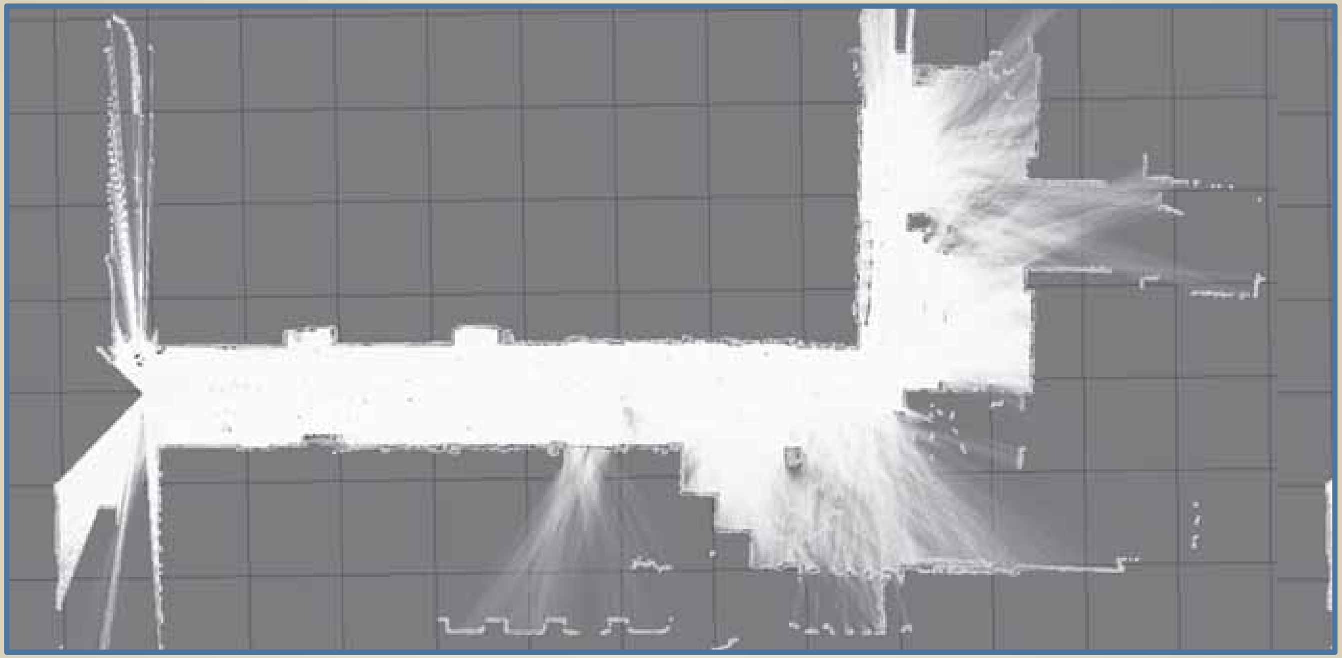
Range data obtained from LIDAR

Result



Point map obtained from 300 frames of range data

Occupancy Grid Map



Converted by Mobile Robot Programming Toolkit, MRPT
http://www.mrpt.org

Accuracy

Indoor & Outdoor

Result by RBPF-SLAM on MRPT

Result by Proposed Method

Conclusion

- Use L0-norm, a kind of robust estimator

- Accelerate by Locality Sensitive Hashing
  → Keep real-time (about 5 fps)

Future work

- Balance accuracy vs. calculation time
  – Cauchy followed by L0-norm
  – L0-norm followed by Biweight
  – L0-norm + moving object detection

etc….

Thank You For Your Attention

Method