# Efficient 2D-Navigation for Mobile Service Robots

Dirk Holz, dirk.holz@ieee.org
Computer Science Department, Bonn-Rhein-Sieg University of Applied Sciences

## Contribution

We propose a set of algorithms for efficiently performing *Simultaneous Localization and Mapping* (SLAM) as well as path planning and motion control. These allow a mobile service robot to efficiently navigate and accomplish complex tasks in dynamic indoor environments.

## SLAM

SLAM is addressed in terms of *incremental range image registration*. The first laser scan $D_0$ makes up an initial point map $M_0$. Points from subsequent scans $D_i, i > 0$ are *matched* against the map $M_{i-1}$. The resulting transformation $\mathbf{T}_i$ is used to correct the robot's pose and the position of the points, yielding the transformed point set $\check{D}_i$. $\check{D}_i$ is then added to $M_{i-1}$ forming $M_i$ to account for new information in $D_i$.
After registering $N$ range scans $D_{i,\ldots,N}$, the point map $M_N$ contains all points measured in the robot's workspace:

$$M_N = \bigcup_{i=[0,N]} \{\check{\mathbf{d}}_{i,j} \mid \check{\mathbf{d}}_{i,j} \in \check{D}_i\}.$$

The key idea of *sparse point maps* is to avoid duplicate storage of points. A point $\check{\mathbf{d}}_{i,j} \in \check{D}_i$ is not added to $M_{i-1}$, if the point-to-point distance to its closest point $\mathbf{m}_{i-1,k} \in M_{i-1}$ is smaller than a minimum allowable distance $\epsilon_{\mathrm{D}}$:

$$M_i = M_{i-1} \cup \{\check{\mathbf{d}}_{i,j} \mid \check{\mathbf{d}}_{i,j} \in D_i, \\ \nexists \mathbf{m}_{i-1,k} \in M_{i-1} : \|\check{\mathbf{d}}_{i,j} - \mathbf{m}_{i-1,k}\| < \epsilon_{\mathrm{D}}\}$$

The transformation $\mathbf{T}_i$ is determined by means of the *Iterative Closest Point* (ICP) algorithm. Given a data set $D$ and a model set $M$, the ICP iteratively searches for $\mathbf{T}_i$ by minimizing:

$$E(\mathbf{R},\mathbf{t}) = \sum_{i=1}^{|M|}\sum_{j=1}^{|D|} w_{i,j}\|\mathbf{m}_i - (\mathbf{R}\mathbf{d}_j + \mathbf{t})\|^2$$

$$w_{i,j} = \begin{cases} 1, & \mathbf{m}_i \text{ corresponds to } \mathbf{d}_j \\ 0, & \text{otherwise.} \end{cases}$$

$$\mathbf{T} = \begin{pmatrix} \mathbf{R}_{ICP} & \mathbf{t}_{ICP} \\ 0\,0\,0 & 1 \end{pmatrix}$$

In the resulting *point maps* the robot can localize itself and other objects.

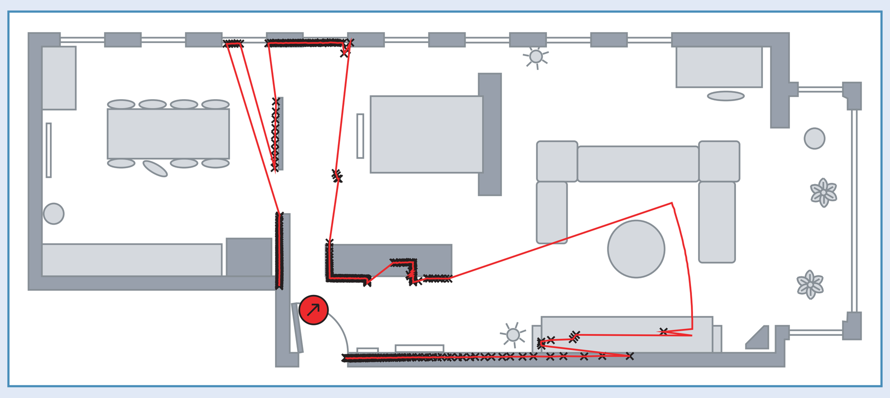
## Path Planning Procedure

1. Construct Voronoi diagram using Fortune's Sweepline algorithm.
2. Construct convex hull using Graham's Scan algorithm.
3. Prune all edges of Voronoi diagram whose distance to neighboring points is smaller than the robot's width and those that lie outside of or intersect the convex hull.
4. Search for shortest path by means of $A^\star$ with cost function

$$f(n) = g(n) + h(n)$$

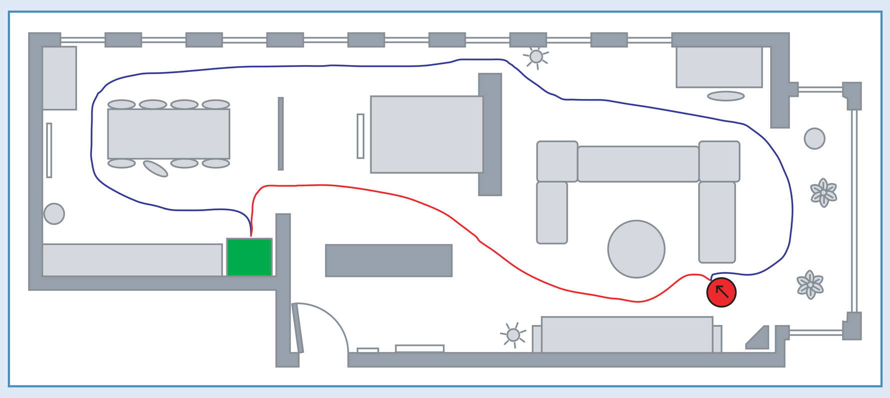
where $g(n)$ is the length of travelled path segments and $h(n)$ the Euclidean distance to the goal.

## Sensor: 2D Laser Scanner

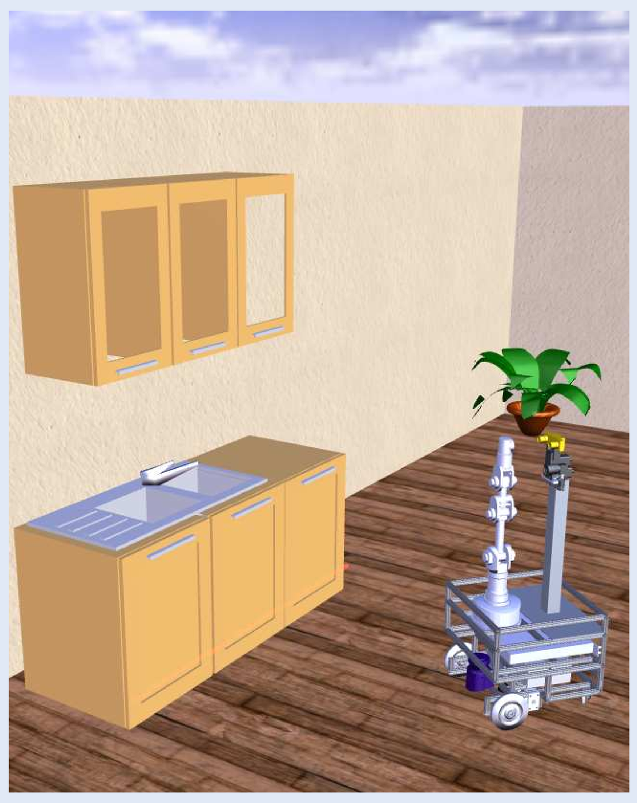


## Typical Applications

"Move to the refridgerator!"



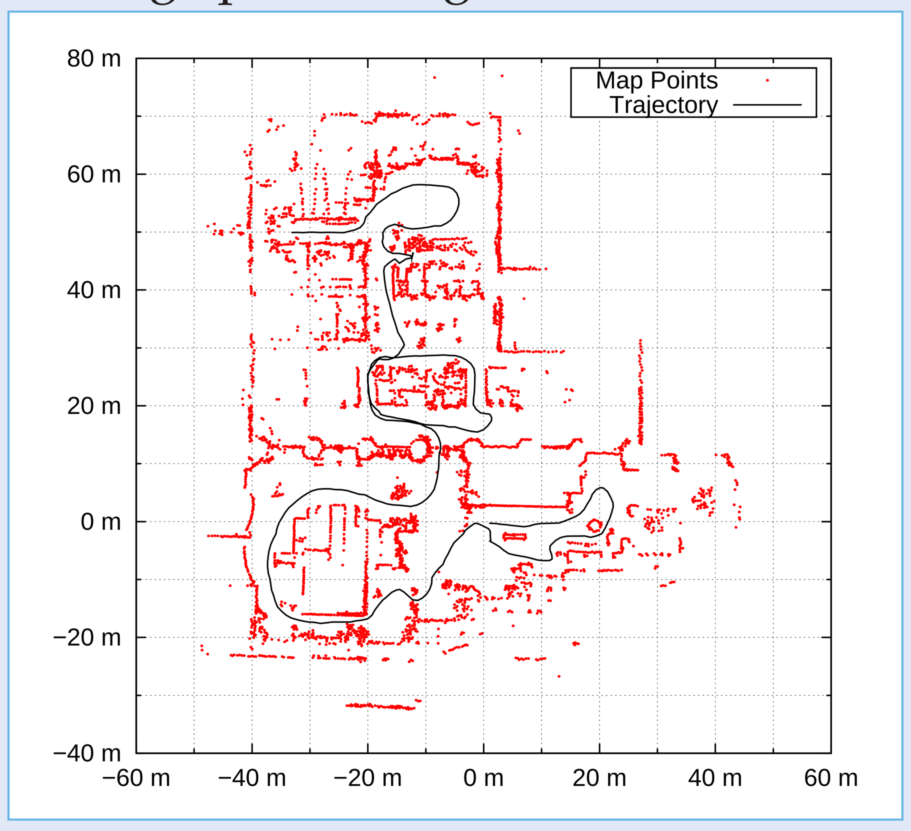
## Platform

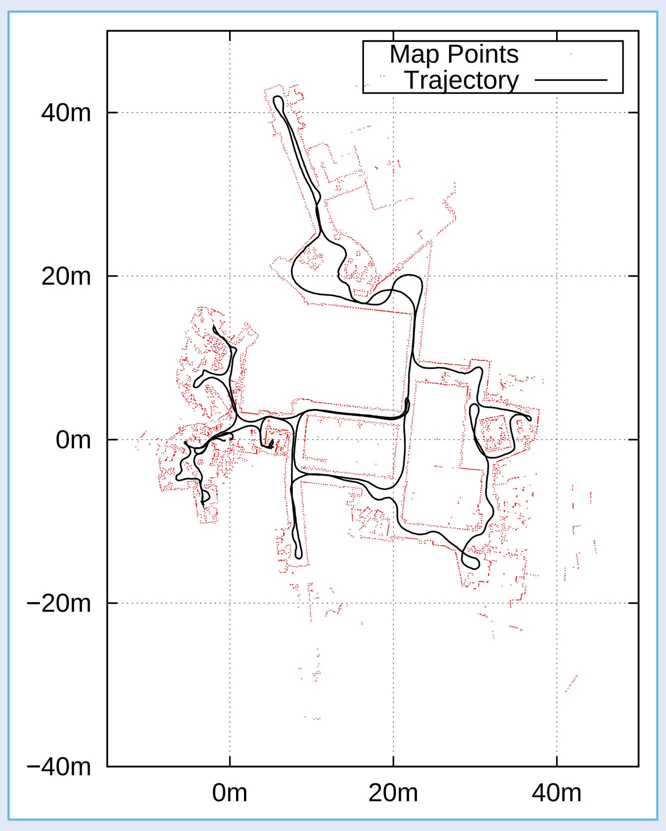


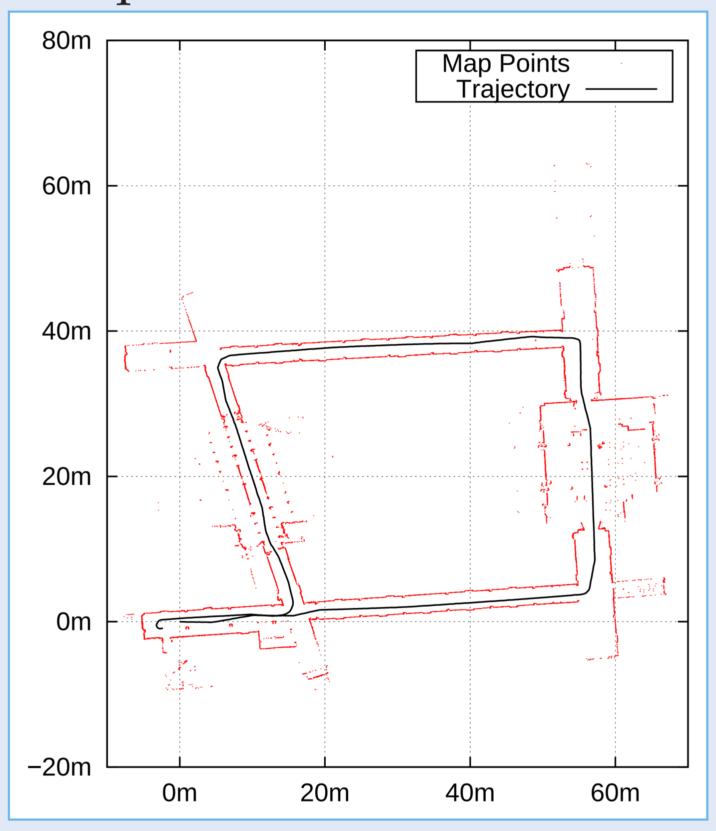
Mobile service robot: Johnny Jackanapes
(Vice-World Champion RoboCup@Home 2008)

## Results: Simultaneous Localization and Mapping

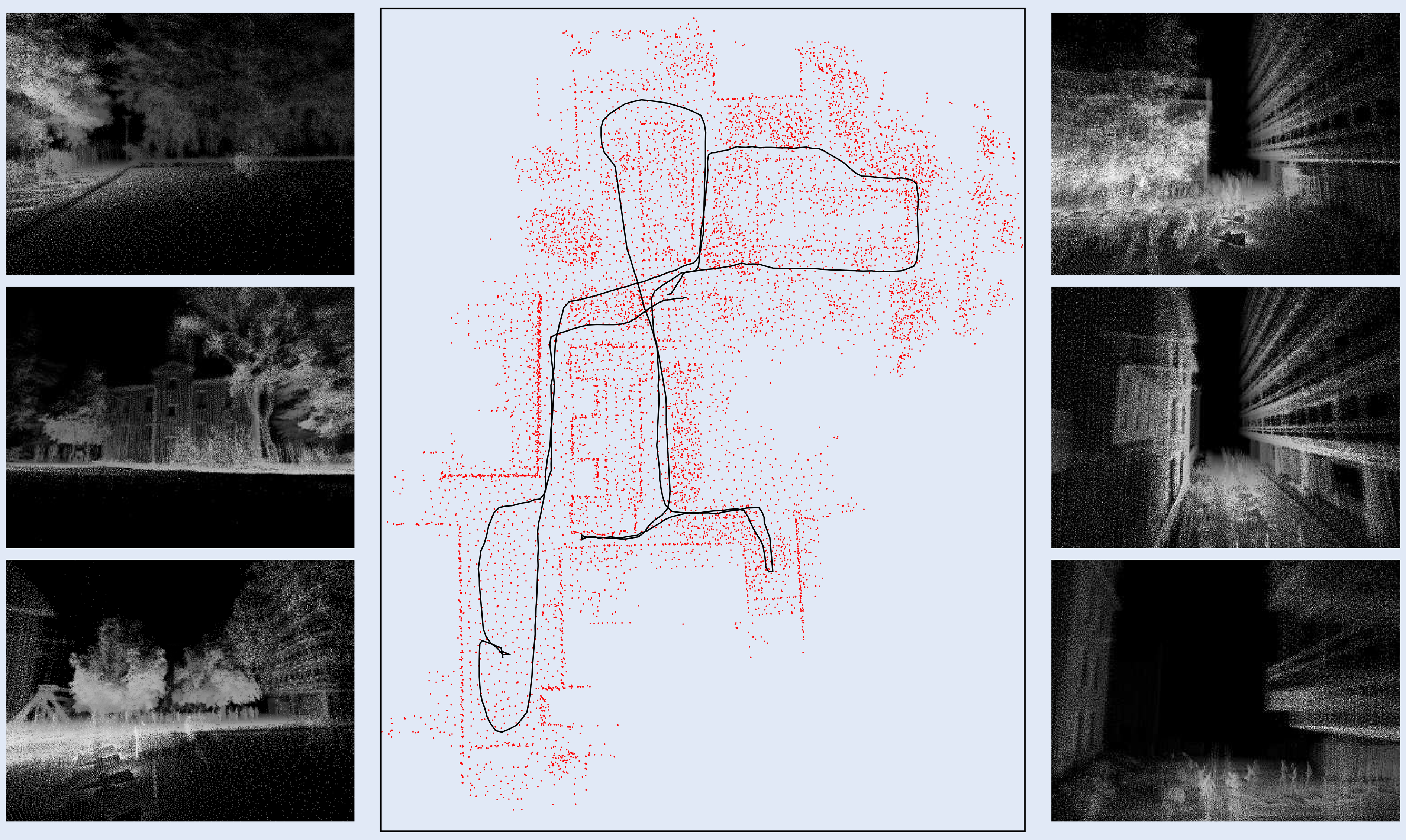
2D Sparse Point Maps ($\epsilon_{\mathrm{D}} = 0.2\,\mathrm{m}$), average processing time per scan: $8\,\mathrm{ms}$
Average processing time with Rao-Blackwellized Particle Filter and 20 particles: $> 150\,\mathrm{ms}$




3D Sparse Point Maps (topview: $\epsilon_{\mathrm{D}} = 2\,\mathrm{m}$, detail views: $\epsilon_{\mathrm{D}} = 0.2\,\mathrm{m}$)
Average processing time per scan: $\approx 35\,\mathrm{ms}$ ($\epsilon_{\mathrm{D}} = 0.2\,\mathrm{m}$), $\approx 450\,\mathrm{ms}$ ($\epsilon_{\mathrm{D}} = 2\,\mathrm{m}$)



## Results: Path Planning and Motion Control

**Initial State:**
Robot stands in the open "FrontDoor"
**Task:** Move to the "Fridge"

Computing Voronoi diagram: $\approx 1.3\,\mathrm{ms}$
Pruning the Voronoi diagram: $\approx 0.8\,\mathrm{ms}$
Planning shortest path (black): $\approx 2.1\,\mathrm{ms}$

Robot successfully followed path and reached target pose by successively applying two non-linear motion conrollers.