Behavior-Tree-Based Person Search for Symbiotic Autonomous Mobile Robot Tasks

Marvin Stüde, Timo Lerche, Martin Alexander Petersen, Svenja Spindeldreier

ICRA 2021

Institute of Mechatronic Systems

Leibniz University Hannover

May 25, 2021

Motivation

Social Service Robots

- Increasing number of applications
- Focus on interaction, rather than manipulation
  - ⇒ Unsolveable situations can occur (doors, lifts)

Figure: Service robots[1]

[1] Top to bottom: SPENCER (KLM), SeRoDI (Fraunhofer IPA), Sobi (imes)

Motivation

Leibniz Universität Hannover

Social Service Robots

- Increasing number of applications
- Focus on interaction, rather than manipulation
  - ⇒ Unsolveable situations can occur (doors, lifts)

Symbiotic Autonomy

Recognize individually unsolveable situations and actively involve humans in problem solving

- Little work on proactive search
- Where to search, where to wait?

Figure: Service robots[1]

[1] Top to bottom: SPENCER (KLM), SeRoDI (Fraunhofer IPA), Sobi (imes)

Approach

Leibniz Universität Hannover

Main Idea

- Utilize Behavior Tree (BT) framework to find people in open spaces
- Synthesize BTs based on a stochastic environmental model of person occurrence

Approach

Main Idea

- Utilize Behavior Tree (BT) framework to find people in open spaces
- Synthesize BTs based on a stochastic environmental model of person occurrence

Why Behavior Trees?

+ Modularity, reusability
+ Problem definition at task level, avoids tailored cost functions
+ Easily extendable with further steps of the symbiotic procedure

Behavior Trees

Behavior Trees

- Directed rooted tree
- Nodes for control flow and action execution

Stochastic Behavior Trees[2]

- Convert BT to Discrete Time Markov Chain
- $\mathcal{A}_{\text{sbt}} : (p_{\text{s}}(t), p_{\text{f}}(t), \mu, \nu)$, $\mathcal{C}_{\text{sbt}} : (p_{\text{s}}(t), p_{\text{f}}(t))$
- Success probability:

$$p_{\text{s,T}}(t) = \sum_{i : s_i \in \mathcal{S}_{\text{S}}} \pi_i(t)$$

- Success rate:

$$\mu_{\text{T}} = \text{avg}\left(\frac{\sum_{i=1}^{|\mathcal{S}_{\text{S}}|} u_{i1}^{\text{S}}(\kappa) \log\left(h_{i1}^{\text{S}}(\kappa)\right)}{\sum_{i=1}^{|\mathcal{S}_{\text{S}}|} u_{i1}^{\text{S}}(\kappa)} \right)^{-1}$$

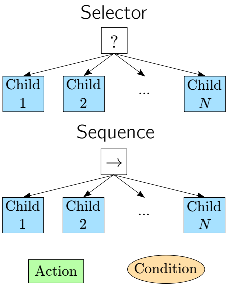

Figure: Notation for different node types

[2] Michele Colledanchise et al. "Performance Analysis of Stochastic Behavior Trees". In: *IEEE ICRA* (2014), pp. 3265–3272.

People Occurrence Model

Poisson process

- Model occurrence rate of people λ in an area as Poisson-distributed
- Probability mass:

$$P\left(N(t) = c\right) = \frac{(\lambda t)^c}{c!} e^{-\lambda t} \quad \text{with} \quad c = 0, 1, 2, \ldots$$

People occurrence model

- Spatial and temporal dependency on the rate $\lambda(\boldsymbol{x}, t)$, $\boldsymbol{x} \in \mathbb{R}^d$
- Approximate λ by Grid $G : \mathbb{R}^{m \times o} \rightarrow \mathbb{R}$ ‡

$$G \ : \quad \lambda(\boldsymbol{x}, t) \simeq \sum_{i=1}^{m} \sum_{j=1}^{o} \lambda_{ij\tau} 1_{ij\tau}(\boldsymbol{x})$$

- Learn each λ_τ incrementally via Bayesian inference with prior $\lambda_\tau \sim \Gamma\left(\lambda_\tau; \alpha_\tau, \beta_\tau\right)$
- α_τ, β_τ depending on robot pose, detection area and number of detected people

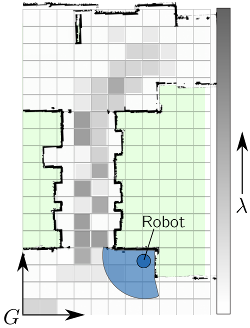

Figure: Schematic illustration of G

‡ Matthias Luber et al. "Place-dependent people tracking". In: *The International Journal of Robotics Research* 30.3 (2011), pp. 280–293.

Behavior-Tree-based Person Search

Goal

- Find a sequence of actions that maximizes the probability of meeting a person

⇒ Where should the robot search, where wait for people?

Behavior-Tree-based Person Search

Leibniz Universität Hannover

Goal

- Find a sequence of actions that maximizes the probability of meeting a person

$\Rightarrow$ Where should the robot search, where wait for people?

Approach

- Define atomic actions $\mathcal{W}_{\mathrm{A},i} \supseteq \mathcal{A}_{\mathrm{sbt}}$ and $\mathcal{S}_{\mathrm{A},i\to j} \supseteq \mathcal{A}_{\mathrm{sbt}}$
  - $\mathcal{W}_{\mathrm{A},i}$: *Wait at place* $\mathcal{P}_i$
  - $\mathcal{S}_{\mathrm{A},i\to j}$: *Search from place* $\mathcal{P}_i$ *to place* $\mathcal{P}_j$

    (both: $\mathcal{P}_i \in G$)
  - Find probabilities to (succeed $p_{\mathrm{s}}(t)$ / fail $p_{\mathrm{f}}(t)$) and rates to (succeed μ / fail ν)
- Find best order of search and wait actions

Wait action definition

Probabilistic parameters

- Success rate

$$\mu_{\mathrm{w}} = \sum_{\mathcal{D}} \lambda_{ij\tau}$$

- Expected time to fail by specifying a confidence p'_{s}
- Fail probability $p_{\mathrm{f,w}}(t; \mu_{\mathrm{w}})$ defined piece wise

Wait action definition

Probabilistic parameters

- Success rate

$$\mu_{\mathrm{w}} = \sum_{\mathcal{D}} \lambda_{ij\tau}$$

- Expected time to fail by specifying a confidence p'_{s}
- Fail probability $p_{\mathrm{f,w}}(t; \mu_{\mathrm{w}})$ defined piece wise

Wait action $\mathcal{W}_{\mathrm{A},i}$ returns...

✓ success, if person is found

× failure, when a maximum time (ν_{w}^{-1}) has been reached

Search action definition

- Search Path: $\mathcal{S}_{i\to j} : \left(\mathcal{P}_i, \mathcal{P}_j, \mathcal{G}, l, \bar{v}\right), \quad i, j \in \{0, 1, \ldots, n\},\ i \neq j$

Probabilistic parameters

- Success rate time dependent

$$\mu_{\text{sp}}(t) = \sum_{\mathcal{D}(t)} \lambda_{ij\mathcal{T}}$$

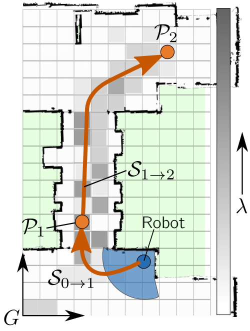

Figure: Illustrative arrangement of search paths

Search action definition

- Search Path: $\mathcal{S}_{i\to j} : \left(\mathcal{P}_i, \mathcal{P}_j, \mathcal{G}, l, \bar{v}\right), \quad i, j \in \{0, 1, \ldots, n\},\ i \neq j$

Probabilistic parameters

⇒ discretizing the path with $t_k = k\,\Delta t$ leads to

$$\mu_{\text{sp,tot}}^{-1} = \underset{t_k \in [t_0, t_0 + l/\bar{v}]}{\arg\min} \left(t_k - \log(1 - p_s')\,\mu_{\text{sp},k}^{-1}\right)$$

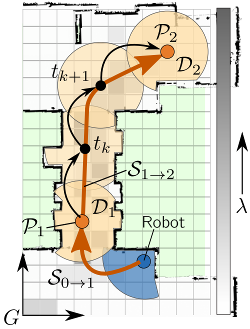

Figure: Illustrative arrangement of search paths

Search action definition

- Search Path: $\mathcal{S}_{i \to j} : \left(\mathcal{P}_i, \mathcal{P}_j, \mathcal{G}, l, \bar{v}\right), \quad i, j \in \{0, 1, \ldots, n\}, \ i \neq j$

Probabilistic parameters

⇒ discretizing the path with $t_k = k\,\Delta t$ leads to

$$\mu_{\mathrm{sp,tot}}^{-1} = \underset{t_k \in [t_0, t_0 + l/\bar{v}]}{\arg\min} \left(t_k - \log(1 - p_{\mathrm{s}}')\,\mu_{\mathrm{sp},k}^{-1}\right)$$

- Fail rate and probability additionally depend on expected distance to fail l_{fail}

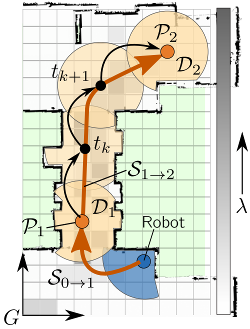

Figure: Illustrative arrangement of search paths

Search action definition

- Search Path: $\mathcal{S}_{i \to j}: \ (\mathcal{P}_i, \mathcal{P}_j, \mathcal{G}, l, \bar{v}), \quad i, j \in \{0, 1, \ldots, n\}, \ i \neq j$

Probabilistic parameters

⇒ discretizing the path with $t_k = k\,\Delta t$ leads to

$$\mu^{-1}_{\mathrm{sp,tot}} = \underset{t_k \in [t_0, t_0 + l/\bar{v}]}{\arg\min} \left(t_k - \log(1 - p_s')\, \mu^{-1}_{\mathrm{sp},k} \right)$$

- Fail rate and probability additionally depend on expected distance to fail l_{fail}

Search action $\mathcal{S}_{\mathrm{A},i \to j}$ returns...

✓ success, if person is found

× failure, when path was driven without finding anyone or navigation fails

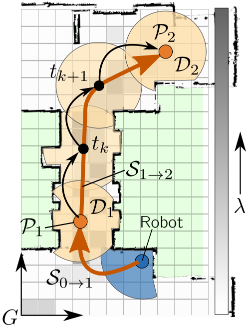

Figure: Illustrative arrangement of search paths

Person Search Behavior Tree (PSBT)

Goal

Find a BT that maximizes the probability of finding a person, taking into account the return time

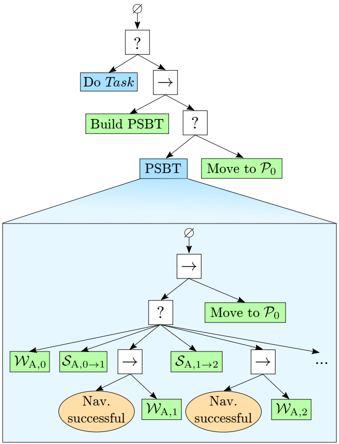

Figure: General form of the PSBT

Person Search Behavior Tree (PSBT)

Goal

Find a BT that maximizes the probability of finding a person, taking into account the return time

Sample n places with probability $p \propto \lambda_{ij}$

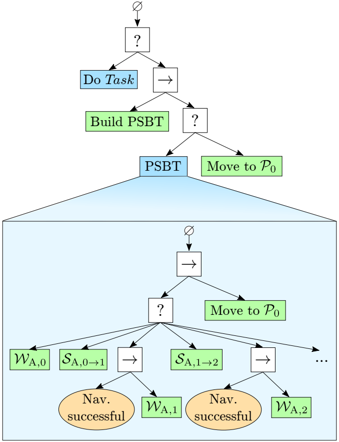

Figure: General form of the PSBT

Person Search Behavior Tree (PSBT)

Goal

Find a BT that maximizes the probability of finding a person, taking into account the return time

Sample n places with probability $p \propto \lambda_{ij}$

↓

Calculate all actions $\mathcal{W}_{A,i}$ and $\mathcal{S}_{A,i\to j}$

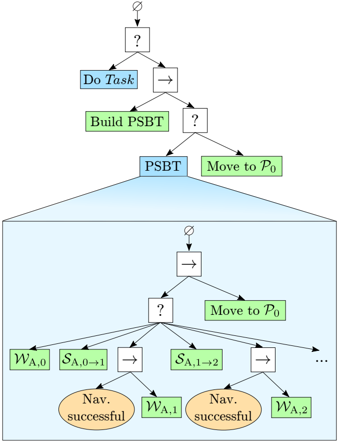

Figure: General form of the PSBT

Person Search Behavior Tree (PSBT)

Leibniz Universität Hannover

Goal

Find a BT that maximizes the probability of finding a person, taking into account the return time

Sample n places with probability $p \propto \lambda_{ij}$

↓

Calculate all actions $\mathcal{W}_{\mathrm{A},i}$ and $\mathcal{S}_{\mathrm{A},i\rightarrow j}$

↓

Create and solve OTSP to reduce complexity

Nodes: $\mathcal{P}_{0\ldots n}$

Costs: $\nu^{-1}_{\mathrm{s},i\rightarrow j}$

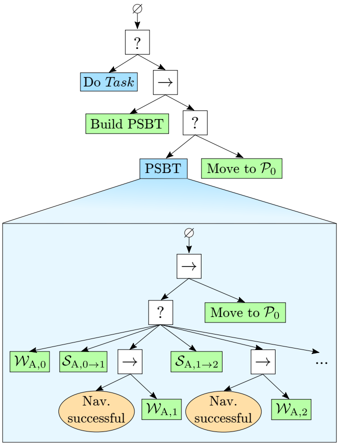

Figure: General form of the PSBT

Person Search Behavior Tree (PSBT)

Leibniz Universität Hannover

Goal

Find a BT that maximizes the probability of finding a person, taking into account the return time

Sample n places with probability $p \propto \lambda_{ij}$

↓

Calculate all actions $\mathcal{W}_{\mathrm{A},i}$ and $\mathcal{S}_{\mathrm{A},i\to j}$

↓

Create and solve OTSP to reduce complexity
Nodes: $\mathcal{P}_{0\ldots n}$
Costs: $\nu_{\mathrm{s},i\to j}^{-1}$

↓

For each possible combination(wait/not wait, search/not search), solve DTMC with generator matrix $Q(\mathcal{A}_{\mathrm{sbt,sp}}, \mathcal{A}_{\mathrm{sbt,w}})$

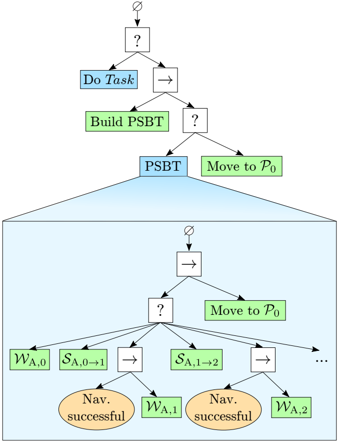

Figure: General form of the PSBT

Person Search Behavior Tree (PSBT)

Leibniz Universität Hannover

Goal

Find a BT that maximizes the probability of finding a person, taking into account the return time

Sample n places with probability $p \propto \lambda_{ij}$

↓

Calculate all actions $\mathcal{W}_{\mathsf{A},i}$ and $\mathcal{S}_{\mathsf{A},i\to j}$

↓

Create and solve OTSP to reduce complexity
Nodes: $\mathcal{P}_{0\ldots n}$
Costs: $\nu^{-1}_{\mathsf{s},i\to j}$

↓

For each possible combination(wait/not wait, search/not search), solve DTMC with generator matrix $Q(\mathcal{A}_{\mathsf{sbt,sp}}, \mathcal{A}_{\mathsf{sbt,w}})$

↓

Choose tree with maximum $p_{\mathsf{s,T}}(t_{\max})$

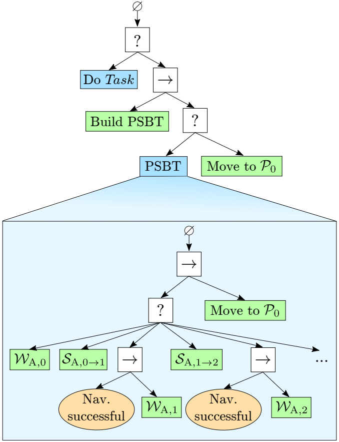

Figure: General form of the PSBT

Environment and Training

Building

- University building with lecture halls, cafeteria, several entrances, sitting areas

Training

- People tracking for two working days
- $\sim 18,000$ people tracks

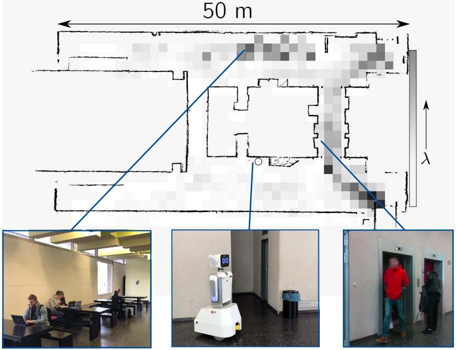

Figure: Environment and learned person occurrence model

Model-based Evaluation

Leibniz Universität Hannover

Experimental setup

- Sample 500 random start locations
- Compare expected success rate, based on person occurrence model

Methods

- **PSBT**: Proposed method
- **GC/GM**: Greedy planning to a (close) cell with large λ
- **W**: Wait at the start location
- **RND**: Random goal sampling
- **NW**: Sample like PSBT, but never wait

Model-based Evaluation

Experimental setup

- Sample 500 random start locations
- Compare expected success rate, based on person occurrence model

Methods

- **PSBT**: Proposed method
- **GC/GM**: Greedy planning to a (close) cell with large λ
- **W**: Wait at the start location
- **RND**: Random goal sampling
- **NW**: Sample like PSBT, but never wait

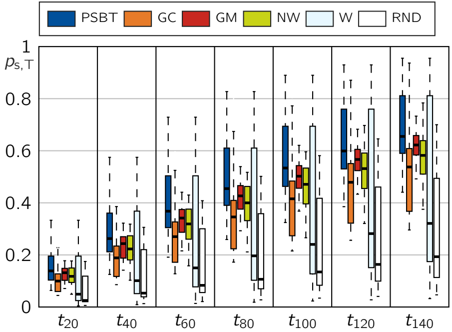

Figure: Probability distribution of the BT root nodes $p_{s,T}(t_k)$ at seven points in time.

Real-world experiments

Experimental setup

- Two start locations, period of five working days
- Online planning and subsequent people search

$\bar{t}_r$: Mean time until person found.

Results

- Total of 588 test runs
- PSBT:
  - Can take longer than greedy methods
  - But: 198 trial runs, 94 % succesful

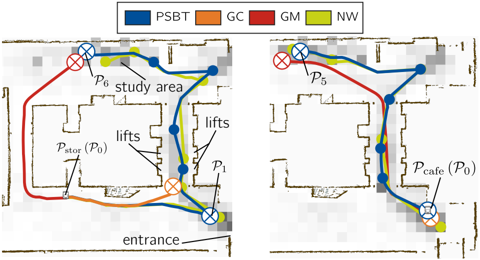

Figure: Exemplary search paths for two different start locations. Crossed circles indicate a waiting location.

Real-world experiments

Experimental setup

- Two start locations, period of five working days
- Online planning and subsequent people search

$\bar{t}_r$: Mean time until person found.

Results

- Total of 588 test runs
- PSBT:
  - Can take longer than greedy methods
  - But: 198 trial runs, 94 % succesful

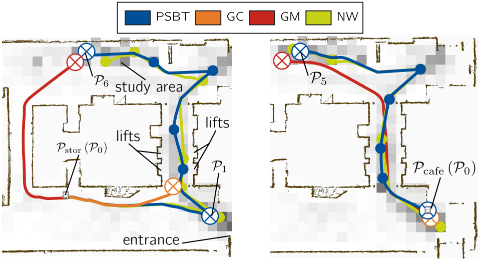

Figure: Exemplary search paths for two different start locations. Crossed circles indicate a waiting location.

Table: Results of the experiments in the university building

	Experimental results				Model estimation
Place	Method	Trials	P	$\bar{t}_r$	$\bar{\mu}_r^{-1}$
$\mathcal{P}_{stor}$	PSBT	86	**98.8 %**	120.5 ± 61.2	139.5 ± 8.7
	NW	93	88.2 %	115.4 ± 52.3	150.8 ± 7.0
	GM	86	62.8 %	101.1 ± 22.4	121.5 ± 0.0
	GC	90	65.6 %	**89.1** ± 49.7	108.1 ± 0.0
$\mathcal{P}_{cafe}$	PSBT	112	**91.1 %**	**35.5** ± 28.2	49.9 ± 0.0
	NW	121	90.1 %	63.0 ± 48.9	126.8 ± 6.0

Summary

- Method for finding people to help problem solving
- Create a Behavior Tree that links wait and search actions
- In 198 trial runs, found people in 94 % of all cases
- Allows for online planning
- Expendable by further steps for symbiotic autonomy