This is the author's version of an article that has been published in the ICRA 2021 proceedings.
Changes were made to this version by the publisher prior to publication.
The final version of record is available at https://doi.org/10.1109/ICRA48506.2021.9561608

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

Marvin Stuede, Timo Lerche, Martin Alexander Petersen and Svenja Spindeldreier[1]

***Abstract*— We consider the problem of people search by a mobile social robot in case of a situation that cannot be solved by the robot alone. Examples are physically opening a closed door or operating an elevator. Based on the Behavior Tree framework, we create a modular and easily extendable action sequence with the goal of finding a person to assist the robot. By decomposing the Behavior Tree as a Discrete Time Markov Chain, we obtain an estimate of the probability and rate of success of the options for action, especially where the robot should wait or search for people. In a real-world experiment, the presented method is compared with other common approaches in a total of 588 test runs over the course of one week, starting at two different locations in a university building. We show our method to be superior to other approaches in terms of success rate and duration until a finding person and returning to the start location.**

I. INTRODUCTION

Social service robots are entering an increasing number of areas of everyday life. Thanks to powerful interaction interfaces, often based on natural language, they are already being used commercially, for example to provide information [1], guiding or automated delivery [2]. Contrary to their strength in socially interacting with people, they usually do not have physical manipulators for reasons of cost and complexity. This prevents them from making full use of the environment and, for example, opening doors [3] or operating elevators without further equipment [4]. An approach to compensate for these weaknesses is *symbiotic autonomy*, which considers the recognition of an individually unsolvable situation and the active involvement of people in problem solving [5]. Depending on the problem, there are often no people available on site to help, so that helpers must be actively searched for. In this paper we utilize the Behavior Tree framework to find people in an open space based on a spatial model of people occurrence rate (see Fig. 1). The method balances between a proactive search and waiting on site to avoid unnecessary travel and waiting times.

Behavior Trees (BTs) are a procedural control approach that has become increasingly popular in the robotics community in recent years. Compared to other approaches, such as hierarchical finite state machines, they have clear advantages in terms of modularity, reusability or expandability [6]–[8]. In addition, they can be analyzed stochastically, which enables a prediction of the probability of success or failure of a particular tree while maintaining the interpretability of the tree's graphical representation [6]. With BTs, in our approach

[1]All authors are with the Leibniz University Hannover, Institute of Mechatronic Systems, D-30823 Garbsen, Germany, marvin.stuede@imes.uni-hannover.de

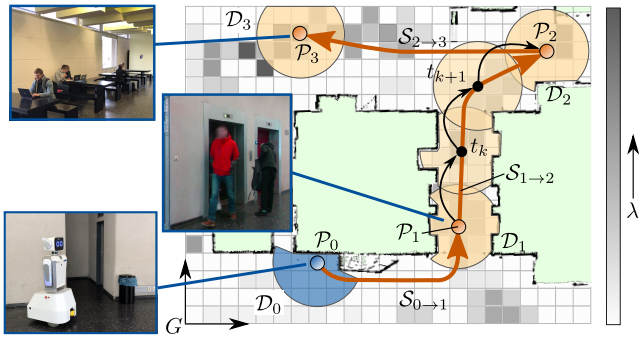


Fig. 1: Exemplary arrangement of different search locations and paths. $\mathcal{P}_i$ denotes a cell in the people occurrence model with rate λ, $\mathcal{D}_i$ a detection zone and $\mathcal{S}_{i\to j}$ a search path. For planning, the detection zone is slid along the search paths with discrete times t_k.

the search for people is modelled flexibly and intuitively. Due to the modularity of the framework, the problem is defined at the level of tasks, such as *wait at a location* or *search along a path*. This avoids the usage of tailored low-level cost functions as used by comparable approaches [9]–[11]. The BTs are synthesized based on a stochastic model, which indicates the occurrence of people using spatially defined Poisson processes. We use the exponentially distributed inter-arrival times of these models as input for a decomposition of the BTs as Discrete Time Markov Chains (DTMCs) for the stochastic analysis. This work explicitly focuses on the search for people and excludes social aspects such as the appropriate approach to a person and verbal questions for help. However, the resulting trees can directly be integrated into a tree that models these aspects without further modification.

II. RELATED WORK

Overcoming the limitations of a robotic system by actively involving humans has been considered in various contexts. Malfunctions can be mitigated by forewarning the users [12] or purposefully utilizing human collaboration in autonomous plans [13]. When a situation occurs where the robot needs help, this must first be identified, for example by detecting a closed door [3] or an elevator [14].

Independent of the search for help, different approaches to finding people were introduced, e.g. based on greedy search [15], hidden Markov Models [9] or Periodic Gaussian Mixture Models [16]. The search for symbiotic autonomous tasks has the additional requirement that people should not only be found, but then accompany the robot to the location where assistance is needed. Since people are only willing to travel a limited distance to help [17], this imposes an additional constraint on the search locations. Most work

the immediate help location, e.g. by supervisors [18] or bystanders [19], and there is only little work on proactively searching and finding people to fulfill a task that cannot be achieved alone by the robot. Rosenthal *et al.* [10] showed that navigating the environment to search for humans could decrease the time until a potential helper is found. They employ an A*-based planner to decide where to seek help in an office building based on the location of offices and availability of the person. However, their method is only evaluated with static locations, using occupancy sensors installed in offices and not applicable to dynamically created locations based on people detections.

Approaches to model the occurrence of people generally use a spatial and temporal partitioning to assign a probability density to each region. Occurrences can then be modeled e.g. temporally via spectral analysis [20] or spatially via direction identification [21]. Another common approach is to use a Poisson process [22] to stochastically model the number and time of occurrences of random events in a time interval. Ihler *et al.* [23] present a non-parametric Bayesian model of intensity functions representing events over time by learning the rate of a Poisson process for a spatially fixed scenario. The authors of [11] extend this approach by spatial and temporal variation of the rate parameter as a piecewise homogeneous Poisson process and then use this model to actively search for people. Although the approach is similar to ours in its theoretical basis, the problem is solved with an individually defined Markov Decision Process (MDP), which is only applicable with lattice-like movement primitives, and difficult to extend and integrate into more advanced models of symbiotic task execution.

In summary, the application of dynamically created people models to symbiotic autonomy and a generally applicable action description to find people for this kind of tasks are open problems. The contributions of this paper are therefore: 1) A task descriptive model of a symbiotic autonomous person search, based on the Behavior Tree framework. 2) The connection of a stochastic environmental model with the BT framework, complying with the requirements of a Stochastic Behavior Tree. 3) Real-world experiments showing the effectiveness of the approach, i.e. in reducing waiting times for spatially problematic helping tasks.

The remainder of this paper is structured as follows: The next section III gives a short overview over BTs and the stochastic people occurrence model. We then introduce how the stochastic actions are derived from the model in Sec. IV. Sec. V experimentally shows the effectiveness of the approach compared to other methods and Sec. VI gives a conclusion.

III. PRELIMINARIES

A. Behavior Trees

A Behavior Tree (BT) is a directed rooted tree, consisting of internal nodes for control flow and leaf nodes for action execution or condition evaluation [8]. Pairs of adjacent nodes are denoted as *parent* (outgoing) and *child* (incoming). The cally sends an enabling signal (*tick*) through the tree that is propagated according to the policies of different control flow nodes. When a node receives a tick, it returns one of three status: *running* (*R*), *success* (*S*) or *failure* (*F*). There are two types of leaf nodes: the action node, which returns *S* if an action was completed successfully and the condition node, which returns *S* according to a pre-defined condition [8].

In the primary model of BTs the execution order of selector and sequence node children is inherently fixed and must be decided beforehand by the designer, which often is non-trivial. To overcome this problem, Stochastic Behavior Trees (SBTs) were introduced by Colledanchise *et al.* [6]. In an SBT each action $\mathcal{A}$ is extended by a tuple $\mathcal{A}_{\mathrm{sbt}} : (p_{\mathrm{s}}(t), p_{\mathrm{f}}(t), \mu, \nu)$ and each condition $\mathcal{C}$ by a tuple $\mathcal{C}_{\mathrm{sbt}} : (p_{\mathrm{s}}(t), p_{\mathrm{f}}(t))$, where $p_{\mathrm{s}}(t)$ $(p_{\mathrm{f}}(t))$ is the probability to succeed (fail) at any given time t. The time to succeed (fail) is a random variable with exponential distribution and rate μ (ν). By describing the inner flow of an SBT as a Discrete Time Markov Chain (DTMC), an indication of the success probability $p_{\mathrm{s,T}}(t)$ of the whole tree can be made by summing up the probabilities of being in one of the DTMC success states $\mathcal{S}_{\mathrm{S}}$:

$$p_{\mathrm{s,T}}(t) = \sum_{i:s_i \in \mathcal{S}_{\mathrm{S}}} \pi_i(t). \tag{1}$$

The probability vector $\pi(t)$ is obtained by solving the Cauchy problem

$$\dot{\pi}(t) = Q(t)\pi(t), \quad \pi(0) = \pi_0, \tag{2}$$

with Q as the infinitesimal generator matrix of the DTMC. The success rate μ_{T} of the tree is calculated as

$$\mu_{\mathrm{T}} = \mathrm{avg}\left(\frac{\sum_{i=1}^{|\mathcal{S}_{\mathrm{S}}|} u_{i1}^{\mathrm{S}}(\kappa) \log\left(h_{i1}^{\mathrm{S}}(\kappa)\right)}{\sum_{i=1}^{|\mathcal{S}_{\mathrm{S}}|} u_{i1}^{\mathrm{S}}(\kappa)} \right)^{-1} \tag{3}$$

with $\mathrm{avg}(.)$ as the average function over time and κ as a time step. The matrices $H^{\mathrm{S}}(\kappa, \boldsymbol{\mathcal{A}}_{\mathrm{sbt}}, \boldsymbol{\mathcal{C}}_{\mathrm{sbt}})$ and $U^{\mathrm{S}}(\kappa, \boldsymbol{\mathcal{A}}_{\mathrm{sbt}}, \boldsymbol{\mathcal{C}}_{\mathrm{sbt}})$ depend on the transit times, number of steps between transient states and success states and the success and failure probabilities and rates of actions $\boldsymbol{\mathcal{A}}_{\mathrm{sbt}}$ and conditions $\boldsymbol{\mathcal{C}}_{\mathrm{sbt}}$. The sets $\boldsymbol{\mathcal{A}}_{\mathrm{sbt}}$ and $\boldsymbol{\mathcal{C}}_{\mathrm{sbt}}$ contain all individual actions and conditions of the tree. The failure probability of the tree $p_{\mathrm{f,T}}(t)$ follows likewise based on the failure states $\mathcal{S}_{\mathrm{F}}$. For more in-depth explanations we refer to [6] and [8].

B. Probabilistic People Occurrence Model

The common approach for probabilistic temporal and spatial description of the occurrence of events, e.g. the appearance of people within an area, is the Poisson process. In general, a Poisson process is a renewal process with Poisson-distributed random variable $(N(t), t \geq 0)$. The probability of $N(t)$ being equal to a count c is given by

$$P\left(N(t) = c\right) = \frac{(\lambda t)^c}{c!} e^{-\lambda t} \quad \text{with} \quad c = 0, 1, 2, ..., \tag{4}$$

where λ is the rate parameter of the process. When $\lambda(t)$ is variable, the process is called *inhomogeneous*. An inhomoge-

dependency $\boldsymbol{x} \in \mathbb{R}^d$ in an Euclidean space $\mathbb{R}^d$ for the rate $\lambda(\boldsymbol{x}, t)$. As proposed in [24], for $\boldsymbol{x} \in \mathbb{R}^2$ the inhomogeneous spatial Poisson process can be approximated by a 2D grid representation

$$G: \quad \lambda(\boldsymbol{x}, t) \simeq \sum_{i=1}^{m} \sum_{j=1}^{o} \lambda_{ij\tau} \mathbf{1}_{ij\tau}(\boldsymbol{x}) \tag{5}$$

with $G : \mathbb{R}^{m \times o} \to \mathbb{R}$, indicator function $\mathbf{1}_{ij\tau}(\boldsymbol{x})$ and $\lambda_{ij\tau}$ as the constant rate of a piecewise homogeneous Poisson process, valid in a time interval $[t_\tau, t_{\tau+1})$.

Learning the probabilistic representation of people occurrences can then be achieved by learning the constant rates $\lambda_{ij\tau}$. For a confidence-sensitive estimation of the rate parameter, Bayesian inference is used with a Gamma-distributed prior $\lambda_\tau \sim \Gamma(\lambda_\tau; \alpha_\tau, \beta_\tau)$ (indices i, j omitted for brevity). The shape parameter α_τ and inverse scale parameter β_τ are determined incrementally. In our case, for a discrete time step σ for all $t_\sigma < t_\tau$ the update rules

$$\alpha_\sigma = \alpha_{\sigma-1} + c_\sigma \, \mathbf{1}_{\mathcal{D}}(\boldsymbol{x}_{\mathrm{R}}, t_\sigma), \quad \beta_\sigma = \beta_{\sigma-1} + \mathbf{1}_{\mathcal{D}}(\boldsymbol{x}_{\mathrm{R}}, t_\sigma) \tag{6}$$

with initial values $\alpha_0 = \beta_0 = 1$ and the number of detected people $c_\sigma \in \mathbb{N}$ since the last time step are used. The indicator function $\mathbf{1}_{\mathcal{D}}(\boldsymbol{x}_{\mathrm{R}}, t_\sigma)$ results from the detection area $\mathcal{D}$ of the robot at pose $\boldsymbol{x}_{\mathrm{R}} \in \mathbb{R}^2$ and causes that only the grid cells which lie within the detection range of the robot will be updated. We use a 3D Lidar for person detection, which provides a full 360-degree environmental view, and therefore approximate $\mathcal{D}$ by a circle with radius r. By providing an estimation of person encounter probability at a specific location, the people occurrence model forms the basis for the decision whether the robot should wait at the place where help is needed or actively search for help.

IV. Behavior Tree based person search

The goal of the BT description is to set up a sequence of actions to maximize the probability of meeting a person, or in other words, to determine if and where the robot should search for or wait for people. To create the tree, the atomic actions $\mathcal{W}_{\mathrm{A},i}$: *Wait at place* $\mathcal{P}_i$ and $\mathcal{S}_{\mathrm{A},i\to j}$: *Search from place* $\mathcal{P}_i$ *to place* $\mathcal{P}_j$ are defined, where a *place* $\mathcal{P}_i \in G$ refers to a specific cell in the people occurrence grid G (eq. 5).

A. Definition of Atomic Actions

For this section we assume that there is a number of n known places the robot could move to and/or wait at, including the robot's position $\mathcal{P}_0$, which is the location where help is needed (Sec. IV-B shows how n is determined). To decide between different behaviors (i.e. Behavior Trees), the tuple $\mathcal{A}_{\mathrm{sbt}}$ must first be defined for each type of action. The wait action $\mathcal{W}_{\mathrm{A},i}$ will return *S* when a person is found and *F* when a maximum time has been reached. The success rate μ_{w} then results directly from the property that the time differences between events of the Poisson process (i.e. interarrival times of people) are exponentially distributed. When the robot starts waiting at a time t_0, the probability

$$f(t; \mu_{\mathrm{w}}) = \begin{cases} \mu_{\mathrm{w}} e^{-\mu_{\mathrm{w}} t} & t \geq t_0 \\ 0 & t < t_0 \end{cases}, \quad \mu_{\mathrm{w}} = \sum_{\mathcal{D}} \lambda_{ij\tau} \tag{7}$$

describes the waiting time. This requires the assumption that $(t - t_0) < (t_{\tau+1} - t_\tau)$ i.e., the rates $\lambda_{ij\tau}$ can be regarded as constant while waiting. The success rate μ_{w} results from the accumulated rates of all visible cells at the waiting position. The probability to meet a person while waiting is given by the corresponding cumulative distribution function

$$p_{\mathrm{s,w}}(t; \mu_{\mathrm{w}}) = \begin{cases} 1 - e^{-\mu_{\mathrm{w}} t} & t \geq t_0 \\ 0 & t < t_0 \end{cases}. \tag{8}$$

By manually specifying a desired confidence $p'_{\mathrm{s}} > p_{\mathrm{s,w}}(\mu_{\mathrm{w}}^{-1}; \mu_{\mathrm{w}})$, this equation can be rearranged to estimate the expected waiting time T' until the next person appears. If no person appears after T', the wait action is considered as failed, resulting in the mean time to fail:

$$\nu_{\mathrm{w}}^{-1} = T' = -\log(1 - p'_{\mathrm{s}}) \, \mu_{\mathrm{w}}^{-1}. \tag{9}$$

Because the wait action will never fail before T' has passed, the fail probability is defined as

$$p_{\mathrm{f,w}}(t; \mu_{\mathrm{w}}) = \begin{cases} 1 - p'_{\mathrm{s}} & t \geq t_0 + T' \\ 0 & t < t_0 + T' \end{cases}. \tag{10}$$

For proactive search, we define a search path as

$$\mathcal{S}_{i\to j} : (\mathcal{P}_i, \mathcal{P}_j, \mathcal{G}, l, \bar{v}), \quad i, j \in \{0, 1, ..., n\}, \, i \neq j, \tag{11}$$

where $\mathcal{P}_i$ and $\mathcal{P}_j$ are the start and end places, $\mathcal{G} \subset \mathbb{R}^2$ is the geometric description, l is the length of the path and $\bar{v}$ the average velocity of the robot while driving on the path. $\mathcal{G}$ and l directly follow from a geometric path planner (such as the A* algorithm on an occupancy grid map) and $\bar{v}$ can either be determined empirically or based on the settings of a local path planner. Similar to the waiting action, we define the searching action $\mathcal{S}_{\mathrm{A},i\to j} = \mathcal{A}_{\mathrm{sbt}} \cup \mathcal{S}_{i\to j}$ to return *S* as soon as a person is found and *F* if the whole path was driven without finding anyone. Additionally, this action can also return *F* if the navigation execution fails, e.g. due to an obstructed goal. While the robot moves on the path, it observes different cells of the people occurrence grid G, each for an individual time span. This imposes a time dependency on the success rate $\mu_{\mathrm{sp}}(t) = \sum_{\mathcal{D}(t)} \lambda_{ij\tau}$ ("sp" denotes *search path*). The rate is calculated by discretizing the path with $t_k = k \, \Delta t$ and calculating the corresponding rate $\mu_{\mathrm{sp},k}$ for each point in time. Fig. 1 illustrates this as an example of different paths in an environment.

The minimal time until a person is found on the path results from the sum of the time t_k and the expected arrival time (as in eq. 9) to

$$\mu_{\mathrm{sp,tot}}^{-1} = \underset{t_k \in [t_0, t_0 + l/\bar{v}]}{\arg\min} \left(t_k - \log(1 - p'_{\mathrm{s}}) \, \mu_{\mathrm{sp},k}^{-1} \right). \tag{12}$$

Here, p'_{s} again is a confidence value and t_0 the time when the search is started. The total success rate $\mu_{\mathrm{sp,tot}}$ of the search path is the inverse of this time. The success probability $p_{\mathrm{s,sp}}(t)$ of finding a person on the path follows as the counter

$$p_{\mathrm{s,sp}}(t;\mu_{\mathrm{sp}}) = 1 - \exp \int_{t_0}^{t} \mu_{\mathrm{sp}}(\tilde{t})\mathrm{d}\tilde{t}. \tag{13}$$

If nobody has been found until the goal is reached or the navigation fails, the action *search person* is considered as failed. The fail rate ν_{sp} is approximated by

$$\nu_{\mathrm{sp}} = \frac{\bar{v}}{l} + \frac{\bar{v}}{l_{\mathrm{fail}}}, \tag{14}$$

where l_{fail} is an expected distance the robot can move until a navigation failure occurs, which we also assume as exponentially distributed. This value can for instance be estimated by observation. Before the proactive search action is finished, the action can only fail due to the navigation and only after it has been finished, due to it not finding a person. The probability $p_{\mathrm{f,sp}}(t;\nu_{\mathrm{sp}})$ to fail is therefore defined as a piecewise function:

$$p_{\mathrm{f,sp}}(t;\nu_{\mathrm{sp}}) = \begin{cases} 1 - p_{\mathrm{s,sp}}(\nu_{\mathrm{sp}}^{-1}), & t \geq \nu_{\mathrm{sp}}^{-1} \\ 1 - \exp(-\frac{\bar{v}}{l_{\mathrm{fail}}}\, t), & t < \nu_{\mathrm{sp}}^{-1} \end{cases}. \tag{15}$$

Introducing l_{fail} into eq. 14 thus increases the probability to fail for longer search paths, giving preference to shorter paths with otherwise equal chance of finding a person. For eq. 15 to be applicable, the condition

$$p_{\mathrm{s,sp}}(\nu_{\mathrm{sp}}^{-1}) \leq \exp(-\frac{l}{l_{\mathrm{fail}} + l}) \tag{16}$$

must hold due to the law of total probability. This is fulfilled for $l \ll l_{\mathrm{fail}}$, which is the case for the present application, since all longer paths can be discarded to avoid searching far away from the help location.

B. Choosing a Behavior Tree

Based on the wait and search actions, an action rule must now be found that maximizes the probability of finding a person, taking into account the return time to the help location. We therefore define the *Person Search Behavior Tree* (PSBT), which describes a sequence of actions that should be executed when the robot is facing a task that cannot be solved by itself. For this, a selector behavior is defined, with a general form as shown in Fig. 2. The tree contains a Move to $\mathcal{P}_0$ action, which describes the movement back to the start location and is interpreted as a search action that can only fail due to the navigation.

To create the PSBT, we first sample a number of n selection, where the probability p of a cell to be selected is $p \propto \lambda_{ij}$. If sampled cells are close to each other (the Euclidean distance is smaller than the detection radius r), only the cell with the larger rate λ_{ij} is kept. Also, only cells with a variance and distance to the robot below specified thresholds can be sampled. Subsequently, all wait actions $\mathcal{W}_{\mathrm{A},i}$ and search paths $\mathcal{S}_{\mathrm{A},i\to j}$ $(\forall i,j \in \{0,1,...,n\},\, i \neq j)$ are calculated. To reduce the complexity $\mathcal{O}(n!)$ of investigating all possible sequences to search all places $\mathcal{P}_{1\ldots n}$, we determine the search order by means of an open traveling salesman problem (OTSP) with places $\mathcal{P}_{0\ldots n}$ as nodes and the inverse of the failure rates $\nu_{\mathrm{s},i\to j}^{-1}$ as costs for the corresponding graph. The OTSP is solved by genetic algorithm [25]. The next step is the stochastic analysis of all possible BTs, with the goal to obtain an estimate of the success probability $p_{\mathrm{s,T}}(t)$ and success rate μ_{T} of the tree (according to eq. 1 and eq. 3) as the basis for decision-making. Each tree's flow is decomposed as a DTMC by solving the Cauchy problem (eq. 2) with the infinitesimal generator matrix $Q(\boldsymbol{\mathcal{A}}_{\mathrm{sbt,sp}}, \boldsymbol{\mathcal{A}}_{\mathrm{sbt,w}})$ until a pre-defined look-ahead time $t_{\max}$. We calculate this for every possible case, namely driving to specific places (or not), waiting there (or not) and finally returning to the help location, leading to a total number of $3^n + 1$ executions. The preferred tree is then chosen as the tree with maximum $p_{\mathrm{s,T}}(t_{\max})$.

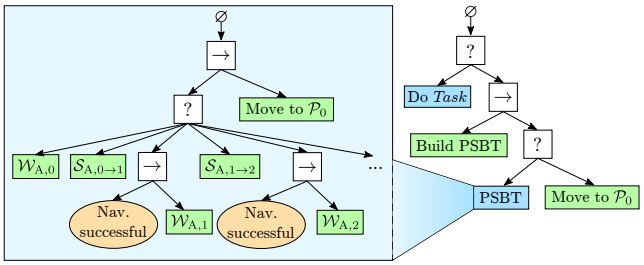


Fig. 2: General form of the PSBT. This tree is executed as a fallback of a $Task$ where the robot needs help, e.g. opening a door or operating an elevator.

V. Experiments

The experimental evaluation is divided into two parts: The evaluation of the performance of the stochastic PSBT based on the people occurrence model and a comparison of the time to find people under real life conditions with the predicted time from the model. The environment for evaluation is the ground floor of a multi-storey 50 m by 25 m university building. The area includes lecture halls, a cafeteria, several entrances and sitting areas (see Fig. 1). For the real-world experiments, the mobile social robot Sobi (see Fig. 3) is used, which is capable of people perception and tracking through 3D Lidar and RGBD cameras. We use an individually trained version of [26] for 3D pointcloud segmentation, extended by the approach proposed in [27] for 2D clustering to decrease the false negative rate. In addition, YOLO v2 [28] is used for detection in the RGB images, which is then registered via a median filter based on the distance either with the 3D

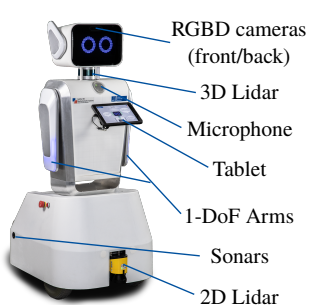


Fig. 3: The ROS-based mobile social robot Sobi.

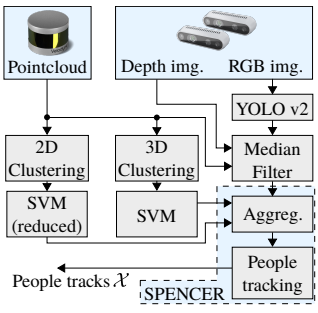


Fig. 4: Person perception and tracking pipeline.

and tracking is performed with the SPENCER framework [29], which outputs people tracks $\mathcal{X}$ in Cartesian space with uncertainty information. Fig. 4 shows an overview of the perception and tracking pipeline. The model is trained with data from two working days by moving the robot throughout the day to different places on the entire floor, so that the same locations are observed at different times of day. Only people tracks with a minimum observed time of 3 s in a range of 5 m are used for training, resulting in a total number of $18,362$ tracks $\mathcal{X}$. This rather high number resides in the fact that the robot partly loses the same person for a short time and then re-detects them as a new track. If a person moves more than one meter or is detected for more than 20 s, this is also considered as a new track. The following parameters were chosen for the experiments: $m = 50$, $o = 25$ (cell resolution of 1 m), detection radius $r = 2\,\mathrm{m}$, confidence $p'_\mathrm{s} = 0.9$, path discretization $\Delta t = 1\,\mathrm{s}$, expected distance $l_\mathrm{fail} = 100\,\mathrm{m}$, average velocity $\bar{v} = 0.5\,\mathrm{m\,s^{-1}}$, sampled goals $n = 6$, DTMC look-ahead time $t_\mathrm{max} = 200\,\mathrm{s}$.

A. PSBT: Model-based Evaluation

We compare the PSBT with five other strategies. The heuristic method of *greedy planning to the global maximum* (GM) plans to the cell with maximum occurrence rate λ and waits there. The method *greedy planning to a close maximum* (GC) is similar to GM, but plans to the closest cell out of the n_λ cells with highest rate and waits there ($n_\lambda = 50$ in this evaluation). *Uniform random sampling of goals* (RND) samples the same number of cells n like PSBT without regarding the rates of the cells and randomly waits at the locations. Furthermore, the trivial approach to *wait at the help location* (W) and the strategy to sample the goals in the same manner as the PSBT, but *never wait* at sampled locations to only search proactively (NW) are implemented.

First, the different methods are compared offline based on the trained people occurrence model. For this, we randomly sample 500 accessible poses on the occupancy grid map of the building as starting poses for the search for helpers. Then the probability of success $p_{\mathrm{s,T}}(t_k)$ is calculated using the methods presented in section IV-A. The comparison is shown in Fig. 5 for seven points in time (every 20 s). Although the observable variance shows a dependency on the starting pose, PSBT outperforms the other methods, leading to an on average 5.7 % higher probability of success after 140 s than the next best method (GM). The PSBT is always at the stochastic analysis also explicitly considers this case and can execute it accordingly. Fig. 6 shows the expected time to success μ_T^{-1} for the four best methods. The time μ_T^{-1} refers to the expected time to find a person and then return to the help location. Here, the expected time for the PSBT is averagely 11.05 s higher than for GM, which is due to the property that the DTMC decomposition yields more conservative estimates of the success rate for an increasing number of selector child nodes. Additionally, the resulting probability distribution $p_{\mathrm{s,T}}(t_k)$ from the DTMC is not exponential, therefore it is not exactly correlated to μ_T^{-1}. Fig. 7 shows two exemplary help locations in front of doors and the corresponding BTs. The location $\mathcal{P}_\mathrm{stor}$ (Figs. 7a, 7b) is in front of a door to a storage room without a large number of people coming and going. While the greedy strategies aim for the closest (or global) maxima, the PSBT strategy moves to the nearby building entrance $\mathcal{P}_1$, waits there and then proceeds to move along the lifts to the global maximum (a sitting area). The second location is one of the entrances to the cafeteria $\mathcal{P}_\mathrm{cafe}$ (Figs. 7c, 7d), which is located in an area with a higher volume of people. Here the PSBT strategy first waits on site before proactively searching for people, while the greedy strategies again aim for the maxima.

B. PSBT: Real-World Experiments

To demonstrate the efficiency of the presented approach, we also conducted real-world experiments. For the two different start locations from Fig. 7, we tested the different methods over a period of five working days. The same constant people occurrence model was used for all methods, which were tested in an alternating fashion in order to compensate for biasing effects (e.g. events in the lecture halls). A single run includes online-planning according to the corresponding method and the subsequent search for people. Instead of approaching the sampled goals directly, the closest reachable pose in a radius of 2 m around the goal is chosen, allowing the sensors to be oriented towards the goal. We consider a person as found if a detected track $\mathcal{X}_i$ of a person is within the radius r in front of the robot for a period of at least 3 s. After a successful search, the robot returns to the starting position and the time for the whole run t_r is saved. A run is considered as failed, when the search tree returns *F*, i.e. when all search and wait actions were executed without finding anyone. Successful searches based on false positive person detections are removed manually from the evaluation.

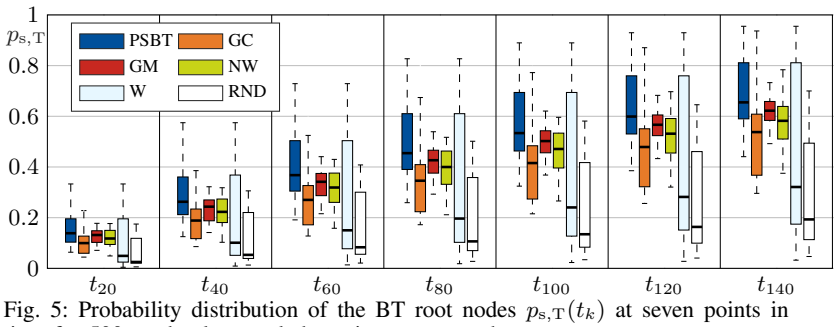


Fig. 5: Probability distribution of the BT root nodes $p_{\mathrm{s,T}}(t_k)$ at seven points in time for 500 randomly sampled starting poses on the map.

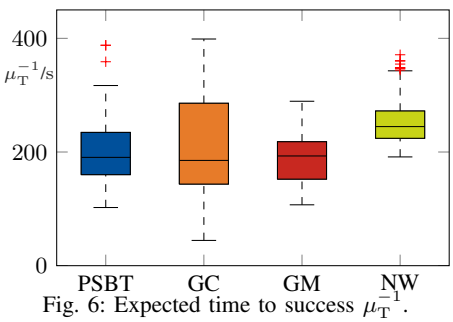
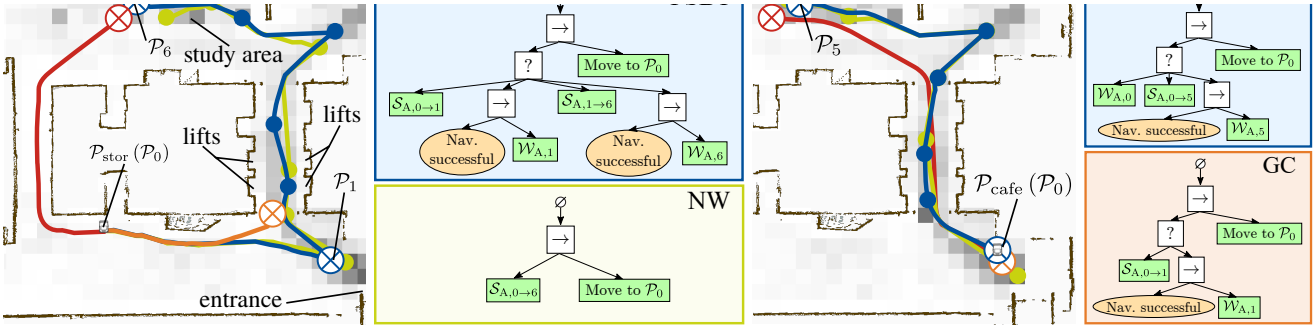


Fig. 6: Expected time to success μ_T^{-1}.

This is the author's version of an article that has been published in the ICRA 2021 proceedings.
Changes were made to this version by the publisher prior to publication.
The final version of record is available at https://doi.org/10.1109/ICRA48506.2021.9561608



(a) (b) (c) (d)

Fig. 7: Exemplary search paths planned by the different methods and corresponding search BTs for a help location $\mathcal{P}_{\text{stor}}$ in an area with low rate λ, and a location $\mathcal{P}_{\text{cafe}}$ with high rate. Colored dots indicate sampled places and crossed circles indicate a waiting location. Consecutive search paths are merged if there are no wait actions inbetween. Colors correspond to the legend in Fig. 5.

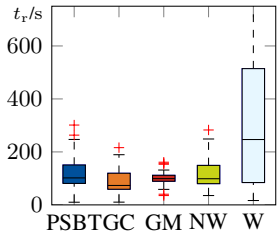


Fig. 8: t_{r} for successful searches from $\mathcal{P}_{\text{stor}}$

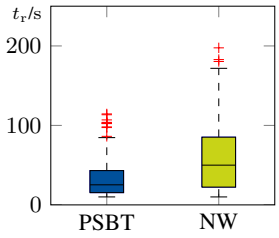


Fig. 9: t_{r} for successful searches from $\mathcal{P}_{\text{cafe}}$

TABLE I: Results of the experiments in the university building with averagely estimated expected time to success $\bar{\mu}_{\text{r}}^{-1}$ and averagely estimated success probability $\bar{p}_{\text{s,T}}(t_{\max})$. Quantitative results given as mean $\pm$ std. deviation.

		Experimental results			Model estimation	
Place	Method	Trials	P	$\bar{t}_{\text{r}}$	$\bar{\mu}_{\text{r}}^{-1}$	$\bar{p}_{\text{s,T}}(t_{\max})$
$\mathcal{P}_{\text{stor}}$	PSBT	86	**98.8 %**	120.5 ± 61.2	139.5 ± 8.7	0.87 ± 0.01
	NW	93	88.2 %	115.4 ± 52.3	150.8 ± 7.0	0.85 ± 0.01
	GM	86	62.8 %	101.1 ± 22.4	121.5 ± 0.0	0.81 ± 0.00
	GC	90	65.6 %	$\mathbf{89.1} \pm 49.7$	108.1 ± 0.0	0.81 ± 0.00
$\mathcal{P}_{\text{cafe}}$	PSBT	112	**91.1 %**	$\mathbf{35.5} \pm 28.2$	49.9 ± 0.0	0.99 ± 0.00
	NW	121	90.1 %	63.0 ± 48.9	126.8 ± 6.0	0.86 ± 0.01

The waiting time for all methods is determined according to eq. 9 with the confidence value p_{s}'. Fig. 8 shows the results for $\mathcal{P}_{\text{stor}}$ as the starting location. All proactive methods perform significantly better than waiting at the help location (W). Table I summarizes the mean time $\bar{t}_{\text{r}}$ with standard deviation, which were calculated for the successful runs only. The shortest average time $\bar{t}_{\text{r}}$ was achieved by the methods GC and GM. Although the greedy methods can be faster, PSBT delivers substantially better results in terms of the success rate P. Since the greedy methods only approach one maximum and remain there, the success depends strongly on whether people can be recognized at this specific location. For example, the global maximum in this environment was a resting area, often occupied by people in slightly different spots, posing larger difficulties for detection on a static waiting location. This is also reflected in the high success rate of the NW method, as it is more likely to detect a person directly in front of the robot during the proactive search. A low success rate is also problematic in that it takes time to determine that all actions have failed (here always more than 200 s), therefore PSBT would still be faster on average compared to GC/GM. An essential advantage of the PSBT method can be seen for highly frequented areas like $\mathcal{P}_{\text{cafe}}$ (see Fig. 9). Here we compare the two best methods from $\mathcal{P}_{\text{stor}}$, to work out the difference between exclusively proactive search and intermittent waiting. The exclusively proactive *never wait* (NW) method takes longer on average to find a person and return to the start location, since PSBT correctly decides that it is more worthwhile to wait on site. Although not evaluated, it should be noted that the greedy strategy GC would lead to similar results as PSBT here (see e.g. the waiting location of GC in Fig. 7c). Based on the expected time to success μ_{r}^{-1} and success probability $p_{\text{s,T}}(t_{\max})$ the model gives an estimate of the performance of the different methods and the necessary time to find a person, thus providing a good estimation for decision making. It provides more conservative estimates for the duration, which is due to the fact that cells with high variance are ignored in the calculation. Furthermore, the assumed average velocity of the robot $\bar{v}$ during the movement, was often exceeded in the test runs. The calculation time of the PSBT averaged 4.8 s for the test runs in this section (Intel i7-7700T CPU), which allows for online execution. To summarize, in 198 trial runs, our method was able to produce suitable predictions of the time until success and found people in 94 % of all cases, which is a higher rate than all other compared methods.

VI. CONCLUSION

We presented a method for finding people in situations where a mobile robot needs human help to solve problems. Based on locally defined Poisson processes to model the occurrence rate of people in an environment, a Behavior Tree is created, linking consecutive search and wait actions to find a person in the shortest possible time. Compared to purely proactive search, the search time can be significantly reduced by actively deciding for or against waiting at the help location while maintaining the success rate. The BT framework enables the modular formulation of the search problem, as well as easy future expandability to include further actions, such as approaching and talking to people or accompanied drives to the help location.

[1] A. K. Pandey and R. Gelin, "A mass-produced sociable humanoid robot: pepper: the first machine of its kind," *IEEE Robotics & Automation Magazine*, vol. 25, no. 3, pp. 40–48, 2018.

[2] S. H. Ivanov, C. Webster, and K. Berezina, "Adoption of robots and service automation by tourism and hospitality companies," *Revista Turismo & Desenvolvimento*, vol. 27, no. 28, pp. 1501–1517, 2017.

[3] M. Stuede, K. Nuelle, S. Tappe, and T. Ortmaier, "Door opening and traversal with an industrial cartesian impedance controlled mobile robot," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 966–972.

[4] F. Wang, G. Chen, and K. Hauser, "Robot button pressing in human environments," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 7173–7180.

[5] M. M. Veloso, J. Biswas, B. Coltin, and S. Rosenthal, "Cobots: Robust symbiotic autonomous mobile service robots." in *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015, p. 4423.

[6] M. Colledanchise, A. Marzinotto, and P. Ögren, "Performance analysis of stochastic behavior trees," *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3265—3272.

[7] C. Paxton, A. Hundt, F. Jonathan, K. Guerin, and G. D. Hager, "Costar: Instructing collaborative robots with behavior trees and vision," in *2017 IEEE international conference on robotics and automation (ICRA)*, pp. 564–571.

[8] M. Colledanchise and P. Ögren, *Behavior Trees in Robotics and AI: An Introduction*. CRC Press, 2018.

[9] P. Bovbel and G. Nejat, "Casper: An Assistive Kitchen Robot to Promote Aging in Place," *Journal of Med Devices*, vol. 8, no. 3, 07 2014.

[10] S. Rosenthal, M. Veloso, and A. K. Dey, "Is Someone in this Office Available to Help Me?" *Journal of Intelligent & Robotic Systems*, vol. 66, no. 1-2, pp. 205–221, 4 2012.

[11] G. D. Tipaldi and K. O. Arras, "I want my coffee hot! learning to find people under spatio-temporal constraints," in *2011 IEEE International Conference on Robotics and Automation*, pp. 1217–1222.

[12] M. K. Lee, S. Kiesler, J. Forlizzi, S. Srinivasa, and P. Rybski, "Gracefully mitigating breakdowns in robotic services," *2010 5th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 203–210.

[13] L. Nardi and L. Iocchi, "Representation and execution of social plans through human-robot collaboration," in *International Conference on Social Robotics*. Springer, 2014, pp. 266–275.

[14] J. Liebner, A. Scheidig, and H.-M. Gross, "Now I Need Help! Passing Doors and Using Elevators as an Assistance Requiring Robot," 2019, pp. 527–537.

[15] M. Volkhardt and H.-M. Gross, "Finding people in home environments with a mobile robot," in *2013 European Conference on Mobile Robots*, pp. 282–287.

"Where's waldo at time t? using spatio-temporal models for mobile robot search," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2140–2146.

[17] S. Rosenthal and M. Veloso, "Mobile robot planning to seek help with spatially-situated tasks," in *Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.

[18] S. Tellex, R. Knepper, A. Li, D. Rus, and N. Roy, "Asking for help using inverse semantics," in *Proceedings of Robotics: Science and Systems*, Berkeley, USA, July 2014.

[19] A. Weiss, J. Igelsböck, M. Tscheligi, A. Bauer, K. Kühnlenz, D. Wollherr, and M. Buss, "Robots asking for directions—the willingness of passers-by to support robots," in *2010 5th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 23–30.

[20] T. Krajnik, J. P. Fentanes, J. M. Santos, and T. Duckett, "FreMEn: Frequency Map Enhancement for Long-Term Mobile Robot Autonomy in Changing Environments," *IEEE Transactions on Robotics*, vol. 33, no. 4, pp. 964–977, 2017.

[21] R. Senanayake and F. Ramos, "Directional Grid Maps: Modeling Multimodal Angular Uncertainty in Dynamic Environments," *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 00, pp. 3241–3248, 2018.

[22] F. Jovan, J. Wyatt, N. Hawes, and T. Krajník, "A poisson-spectral model for modelling temporal patterns in human data observed by a robot," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4013–4018.

[23] A. T. Ihler and P. Smyth, "Learning time-intensity profiles of human activity using non-parametric Bayesian models," in *Advances in Neural Information Processing Systems*, 2007, pp. 625–632.

[24] M. Luber, G. D. Tipaldi, and K. O. Arras, "Place-dependent people tracking," *The International Journal of Robotics Research*, vol. 30, no. 3, pp. 280–293, 2011.

[25] D. Davendra, *Traveling Salesman Problem: Theory and Applications*. InTech, 2010.

[26] Z. Yan, T. Duckett, and N. Bellotto, "Online learning for human classification in 3D LiDAR-based tracking," *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 864–871.

[27] I. Bogoslavskyi and C. Stachniss, "Fast range image-based segmentation of sparse 3D laser scans for online operation," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 163–169.

[28] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7263–7271.

[29] T. Linder, S. Breuers, B. Leibe, and K. O. Arras, "On multi-modal people tracking from mobile platforms in very crowded and dynamic environments," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5512–5519.