Bootstrapping Navigation and Path Planning Using Human Positional Traces

Alen Alempijevic, Robert Fitch and Nathan Kirchner

***Abstract*— Navigating and path planning in environments with limited *a priori* knowledge is a fundamental challenge for mobile robots. Robots operating in human-occupied environments must also respect sociocontextual boundaries such as personal workspaces. There is a need for robots to be able to navigate in such environments without having to explore and build an intricate representation of the world. In this paper, a method for supplementing directly observed environmental information with indirect observations of occupied space is presented. The proposed approach enables the online inclusion of novel human positional traces and environment information into a probabilistic framework for path planning. Encapsulation of sociocontextual information, such as identifying areas that people tend to use to move through the environment, is inherently achieved without supervised learning or labelling. Our method bootstraps navigation with indirectly observed sensor data, and leverages the flexibility of the Gaussian process (GP) for producing a navigational map that sampling based path planers such as Probabilistic Roadmaps (PRM) can effectively utilise. Empirical results on a mobile platform demonstrate that a robot can efficiently and socially-appropriately reach a desired goal by exploiting the navigational map in our Bayesian statistical framework.**

I. INTRODUCTION

Whilst capabilities of robots to perceive and interact with the world are maturing, dynamic environments where robots are deployed with limited *a priori* knowledge still pose significant challenges. These challenges arise during simultaneously localising and mapping under uncertainty, exploring the unknown environment, and creating a suitable navigational representation for path planning. Thus far, a common approach is to employ a method targeted at minimising the impact of such challenges directly. This can be done through using mapping sensors which are unaffected by moving objects, such as a camera which observes the ceiling [1], or by jointly tracking and localising to build an occupancy grid representation [2]. Alternatively, planning trajectories with an incomplete navigational representation can be achieved with a purely reactive planner based on potential fields [3]. However, these approaches under utilise the available sensor information, are prone to local minima and require hand tuning of objective functions.

This work is supported by RobotAssist and the Centre for Autonomous Systems, University of Technology, Sydney.
Alen Alempijevic and Nathan Kirchner are with the Centre for Autonomous Systems, University of Technology Sydney, Sydney, Australia. {Alen.Alempijevic, Nathan.Kirchner}@uts.edu.au.
Robert Fitch is with the Australian Centre for Field Robotics, University of Sydney, Sydney, Australia. r.fitch@acfr.usyd.edu.au

As robots start to share the environment with humans, the analysis of human motion patterns becomes increasingly important in order to improve interaction through predictability. Commonly the trajectory a person will take is estimated [4], [5] or the final goal is determined [6]. However, trajectories taken by people are a product of intention. By learning how humans move in indoor environments a robot can actively navigate without relying solely on direct perception. For example, people generally avoid obstacles or potential hazards like spilt liquids. Trying to sense this is difficult, yet inferring it from the cues of people walking is relatively easier. Additionally, human locomotion provides insight into rich sociocontextual information. For instance, social boundaries such as personal workspaces or the area between a television and a viewer often mean that traversing certain areas is not desirable. Integrating such information into a path planner is nontrivial; definitions of socially acceptable routes are mostly qualitative and difficult to sense [7]. However, examining motion patterns can produce trajectories that incorporate higher levels of reasoning about the environment without explicitly modelling underlying principles influencing them. Thus, incorporating generalised human behaviour, in the form of a probabilistic representation of robot-sensed human positional traces, can enable robots to construct a significantly richer understanding of the environment and its intricacies.

Incorporating human motion patterns effectively in a probabilistic representation has been attempted with a hierarchical Bayesian Model [8] or employing a Hidden Markov Model [9], [10]. Similarly, collections of trajectories characterizing motion patterns have been investigated in [10] to prevent collisions between a robot and person. Ikeda et. al [6] examined the concept of grouping trajectories by the use of sub-goals where pedestrians take directional choices before reaching a final destination. This body of work has demonstrated that it is possible to estimate the most probable trajectory a person will undertake. However, these approaches require a map (occupancy grid) for path planning and/or substantial *a priori* environment knowledge to be able to ascertain viable goals. Our approach is distinct in that we seek to exploit human trajectory observations without the need to make assertions about their final destination. We aim to exploit all available motion patterns to extrapolate a navigational map representation.

In the absence of *a priori* map representation for navigation, current state of the art exploration strategies produce a map by pursuing frontiers [11]. As robots share the environment with humans, giving instructions to the robot such

as bearing only direction of travel has also been investigated [12]. Rather than relying on human input to determine the direction of travel we advocate that equipping the robot with person detection capabilities *in situ* will produce indirect observations that can be exploited for effective path planning. A navigational map constructed without any prior environment information was examined in our prior work [13]. Gaussian Processes (GP) were used to learn a navigational function that describes how human motion deviates from a shortest path prior. Whilst fundamentally complete, this approach can only utilise positional traces that coincide with the intended goal of the robot and can only be effectively combined with a reactive path planner.

This paper presents a method of building a navigational map environment representation utilising human positional traces. What differentiates our approach from existing approaches is utilisation of all available positional traces that do not necessarily coincide with our final goal. The representation can then be exploited by standard planning algorithms (such as PRM) for online path planning. This allows the representation to be updated as either more positional traces become available or more of the environment is perceived, thus, refining the path planning whilst the platform is already in motion. We demonstrate that our approach has significant advantages over standard exploration which is unable to explicitly encode desirably avoidable trajectories. Finally, a key aspect of our approach is the encoding of sociocontextual information from the navigational map, such as identifying areas that people tend to use to move through the socially-acceptable environments (corridors for instance) and avoid the socially-unacceptable environments (someone's cubicle for instance) without any supervised learning or labelling.

The breakdown of this paper is as follows: Section II details our method for extracting human positional traces, Section III discusses the method of encapsulation of this information effectively using a GP and bootstrapping a sampling based path planner to this representation. An empirical evaluation of our approach is presented in Section IV. Finally, conclusions are drawn and future work proposed in Section V.

II. HUMAN POSITIONAL TRACES

As can be appreciated from the aforementioned, the ability to construct human positional traces is an essential prerequisite of our proposed system. This prerequisite comprises of two primary components of Person Detection and Person Tracking, both of which are detailed in the following subsections.

A. Person Detection

The first stage of our system robustly detects people in 3D pointclouds (from a Microsoft Kinect) or laser rangefinder (LRF) data (from a Hokuyo UTM-30LX). As this stage is built upon the authors' previous work [14], [15], [16] an overview will be presented here, for the sake of completeness, and the reader is referred to our original publications for full details.

The Person Detection process commences with projecting the 3D pointcloud data onto a 2D horizontal plane via a bivariate histogram. Blob Detection is then performed on the image to allow each apparent person (vertical surface of appropriate width) to be segmented from the scene. This approach to scene analysis exploits the assumption that people appear in a scene as vertical surfaces with a relatively high point-density in the histogram, when compared with the many horizontal surfaces (floor, chairs, etc.). Along with humans, common items identified as potentially human include: walls, doors, and tall items of furniture. Many of these false positives are eliminated by liberal object-of-interest size constraints based on the expected minimum and maximum size of a person. However, as shown in [14], for the sake of robustness, false positives passing this stage of the detection process are preferable to false negatives as further discrimination is performed using a more discriminative, but more computationally expensive, method. Specifically, a scale and viewing angle robust feature vector (Head-to-shoulder signature - HSS) [15] is constructed. Person detection is then subsequently achieved via a single class support vector machine (SVM) trained (supervised) with HSS signatures guaranteed to represent a range of people of diverse sizes and shapes.

The algorithm for people detection on 2D LRF data is based on extraction of the human torso [16]. The LRF mounted on our mobile robotic platform is at approximately 1.3m above the ground and though laser scans intersect the chest section of a person the method used for torso extraction was directly applied. LRF data is segmented based on discontinuities, major and minor axis of each data segments is extracted and classified using a (SVM) trained (supervised) guaranteed to represent a range of people of diverse sizes and shapes. The outputs of these two methods are statistically combined.

B. Person Tracking

Having detected people a particle filter is used in order to perform global tracking and produce positional traces. To estimate each positional trace a set of samples $X_t = \langle x_t^i \mid i = 1...N \rangle$ and its associated weights ω_t^i represent the beleif at time t of the persons location. The computation of the posterior for each t-th particle set X_t is then calculated recursively from X_{t-1} in three steps as detailed in Alg.1. A constant velocity model driven by zero mean gaussian noise u_t is used in the prediction step. Due to the resampling the particle filter tends to converge to one state, which means that in the basic implementation this filter would not be suitable to track multiple hypotheses over extended periods of time. However, as our method exploits all positional traces as a whole, this is non-problematic and the partial traces are incorporated into the navigational map without the need for more sophisticated tracking.

III. GAUSSIAN PROCESS NAVIGATIONAL MAP

The positional traces are used to produce a navigational map encoded with a GP and this map can be updated as

Algorithm 1 The three steps of the particle filter algorithm

1: **Prediction:** Draw $x_t^i \sim p(x_t^i \mid x_{t-1}^i, u_{t-1})$.
2: **Update:** Compute the weights $\omega_t^i = \eta p(y_t \mid x_t^i)$, with η being a normalization factor to ensure that the weights sum up to 1. Here, y_t is a person detection reading at time t.
3: **Resample:** Draw new x_t^i using weights w_t^i

either more positional traces become available or more of the environment is perceived. Finally, the Path Planning stage exploits the Bayesian formulation of the GP to construct a trajectory which seeks to maximize the most commonly traversed space. Each of these stages are detailed in the following sub-sections.

A. Navigational Map

The work of GP maps was introduced by O'Callaghan et al. [17] who employ Gaussian Process as a Probabilistic Least Square Classifier for reconstruction of a 2D occupancy based map using sparse LRF beam information. Gaussian Process are nonparametric tools for regression and provide a powerful framework for learning models of spatially correlated and uncertain data. GPs represent a family of distribution over functions and inference takes place directly in the function space. Intuitively, this approach to mapping exploits the fact that environments contain spatial structure to predict a continuous non-linear, non-parametric function representing the map. The GP is a Bayesian regression technique and intrinsically avoids overfitting while using statistical inference to learn dependencies between points in the input (training) dataset. A detailed explanation and derivation of the GP can be found in [18].

We base our method on the ability of the GP to predict $p(\Psi|x)$ where Ψ is the nonparametric model of the occupancy at position x in $\mathbb{R}^2$. The GP is used to fit a likelihood function to training data $X = (x_i, y_i)|i = 1, ..., n$, x_i is the two dimensional training input (a single position of a person along the positional trace) and y_i the corresponding scalar training output in our case, y_i is +1 or 0 (for occupied and non-occupied respectively). The resulting continuous function can then be used to predict the probability of occupancy over the entire region. By assuming that all points in the navigational space are jointly Gaussian we obtain

$$\Psi(x_*) = \mathcal{N}(\mu, \sigma^2) \tag{1}$$

where the mean function μ and covariance function $K(x, x_*)$ are

$$\begin{gathered}\mu = k(x_*, X)^T[k(X, X) + \sigma_n^2 I]^{-1} y, \\ \sigma^2 = k(x_*, x_*) - k(x_*, X)[k(X, X) + {\sigma_n}^2 I]^{-1} k(X, x_*)\end{gathered} \tag{2}$$

x_* refers to a query location, X is training inputs, ${\sigma_n}^2$ is the variance of the noise and k is the covariance matrix with elements $k_{ij} = k(x_i, x_j)$ defined on the covariance function k hyperparameters.

After testing with several covariance functions and considering the input variables x_i which are close in input space are highly correlated whilst those far away are uncorrelated, the Squared Exponential covariance function k was chosen. In particular the Squared Exponential covariance function with Automatic Relevance Determination (ARD) [18] distance measure was selected. The covariance function is parameterized as

$$k(x^p, x^q) = {\sigma_f}^2 \exp(-(x^p - x^q)' M x^p - x^q 2{\sigma_n}^2 \tag{3}$$

where the M matrix is diagonal with ARD parameters $l_1^2, ..., l_D^2$, where D is the dimension of the input space and ${\sigma_f}^2$ is the signal variance and ${\sigma_n}^2$ the optional noise level parameter

The training and updating phase of the GP can run concurrently as per Alg.2. To start this process we generate training samples X each with occupancy $y = 0.5$ representing unknown space and learn hyperparameters Θ of underlying function Ψ. Thereafter as either more positional traces are detected or more obstacles in the environment are perceived we update the GP according to Alg.2.

Algorithm 2 Learning Mechanism

Input: X - training samples, Θ - hyperparameters, X_* - positional traces of size N, ${\sigma_{min}}^2$ - min acceptable uncertainty
Output: Θ - parameters of Ψ, X - training samples
Obtain μ, σ^2 for X_* (query locations) from $\Psi(\Theta)$ via Eq.2
for $i = 1$ to N **do**
  if $\sigma^2 < \sigma_{min}$ **then**
    $\hat{X} = (x_i, y_i)$ where $y_i = 1.0 + \eta$ presents occupied space or $y_i = 0.0 + \eta$ presents free space, η is zero mean gaussian noise
  end if
end for
if number of elements in $\hat{X} > 0$ **then**
  concatenate previous accumulated data X and $\hat{X}$
  Train Gaussian process with X to learn new parameters Θ of underlying function Ψ
end if

The primary drawback of using the Gaussian Process is its typical computational time, which makes it unsuitable for many operations where realtime performance are required. The GPs runtime complexity is of $O(N^3)$ (N is the number of training points) and is primarily a result of the requirement to invert the covariance matrix K in eq 2. However, as this matrix is independent of query points the inverse K^{-1} can be pre-computed and stored thus eliminating the computational bottleneck during online use. With additional observations of either positional traces or occupied space we need to retrain the GP. Some work on subsuming this information into K^{-1} has been demonstrated in [19]. Our approach is to handle the computational complexity by storing the inverse covariance matrix K^{-1} for querying and implementing training

the covariance function using cloud computing capabilities. Further details on computational resources and inter process communication specific to our implementation are discussed in Sec.IV.

B. Path Planning

The resulting predictive mean and variance distributions resulting from queries on the GP can be used to distinguish areas in the robot's surroundings where people commonly traverse. The continuous nature of the underlying function means that the navigational map can be queried at an arbitrary coarseness. In such, the GP navigational map allows the inference of the probability of occupied space for any query point in continuous $\mathbb{R}^2$ space. Whereas, path planners utilising traditional occupancy grid based maps are limited by the resolution of the grid size. Thus, this continuous property maximizes the flexibility of path planner selection; any standard sampling based algorithm can be employed. The traditional metric for path selection is based on the shortest distance travelled, a more suitable objective functions such as maximizing the most traversed space is obtained from the GP 2 and can be incorporated in a Rapidly-exploring Random Trees (RRT) or a Probabilistic Roadmap (PRM) planner.

The PRM planner procedure is presented in Alg.3 where the inputs consist of the start pose S goal pose G, maximum probability of occupancy for a valid node p_{max}, maximum number of nearest nodes to consider k and δ distance along each edge to query occupied space in underlying GP.

Algorithm 3 Feature Extraction

Input: S, G, p_{max}, k, δ
Output: Probabilistic Roadmap Network - PRM, Nodes to visit N
while G not reached and PRM not connected **do**
  create N random samples (nodes) X and perform inference on $\Psi(\Theta)$ Eq.2
  for $j = 1$ to (N) **do**
    if X_i not in collision $\mu < p_{max}$ **then**
      Find k nearest nodes
      Collision check along each edge at every δ
      Determine a edge weight according to the probability of occupancy along each edge
    end if
  end for
  Search graph to extract PRM with smallest edge weight, save list of nodes visited N
end while

The nodes N produced by searching the PRM are then passed on to a path planner that can either fit splines to smooth the paths with kinematic constraints of the platform or perform a reactive collision avoidance method. Further details on our path planner and the switching between these two modalities of path planning is discussed in [20].

IV. RESULTS

This section presents results from an empirical evaluation conducted at the Centre for Autonomous Systems (CAS), University of Technology, Sydney 2. The research platform and software framework are briefly presented, the environmental setting and results are then discussed.

Fig. 1. The RobotAssist Platform

Our research platform, RobotAssist, is shown in Figure 1 and is based on a Segway RMP primarily instrumented with an XSens IMU, two Hokuyo UTM-30LX LRF, an RGB-D camera (the SwissRanger/Dragonfly combination has been replaced with a Microsoft Kinect) and an articulated head (2DOF, pan and tilt). The intention is to allow the sensors in the head to be moved quasi-independently from the platform in order to facilitate active sensing. Our platform stands approximately 1.6m tall and weighs approximately 70kg, additional details are available from [21] and *www.robotassist.org*.

Our software is built using a Component-Based Software Engineering (CBSE) paradigm, components run asynchronously and exchange information. We use ZeroC's Ice middleware (`http://www.zeroc.com`) extensively in our system for component interface definition, inter-component communication, component deployment, location, activation services, etc. The multi OS and multi programming language support of Ice was crucial for the development of the robotic system, enabling the development of closed loop control with use of real-time acquired data from the platform in MATLAB environments. The GP Navigational Map is constructed and queried using MATLAB on the High Performance Cluster (by virtue of cloud computing) containing an Intel Xeon X5690 (6 Core) @ 3.73GHz with 96GB DDR3-RAM. The cluster is particularly suited to performing large parametric searches and to algorithms that lend themselves to parallelisation such as the GP presented here within.

The experiments were conducted with our person recognition system configured to operate using the Microsoft Kinect and LRF sensor on the RobotAssist platform [21]. The

Fig. 2. The Experiment Setting: the CAS Office space is densely populated

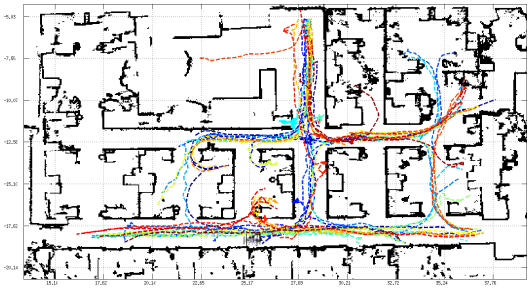

Fig. 3. All positional traces acquired superimposed on a previously acquired map of the CAS area

experiment was conducted in CAS office-space area where 50 students and staff share a space of approximately $50m^2$. A section of the experiment setting can be seen in Fig. 2.

A. Stationary Observer Navigational Map Creation

The first experiment was conducted with the RobotAssist platform stationary, observing human motion over 140 minutes during normal work hours. A total of 512 raw positional traces were collected over this time, all the positional traces are denoted in Fig. 3. The resulting GP learning phase has produced a navigational map Fig. 4 where $p = 1.0$ (indicated in *red*) relates to areas determined to be obstacle free. The associated covariance is presented in Fig. 5.

The more frequently traversed corridors have a lower level of uncertainty on the predicted value of occupancy. The sociocontextual information, such as recommending to avoid cubicles where the flow of pedestrian traffic can not be detected is embedded in the navigational map.

B. Path Planning on the Navigational Map

To demonstrate that the robot can continuously query and update a Navigational Map we conduct an experiment where the mobile platform is simultaneously performing detection, tracking, producing and updating the navigational map, path planning and executing a trajectory. The probability of occupancy for each cell has been updated during the execution of the planned trajectory resulting and the final resulting navigational map is denoted in Fig. 6 and Fig. 7. The information of obstacles encountered (walls) is visible by examining areas that have a probability of being free space as $p = 0.0$ (noted in *blue*) on Fig. 6. The resulting navigational space where the probability of free space is $p > 0.9$ is superimposed on a previously acquired map of the area Fig. 8 for comparison. The path recommended for traversing computed by the PRM is denoted in Fig. 9 while

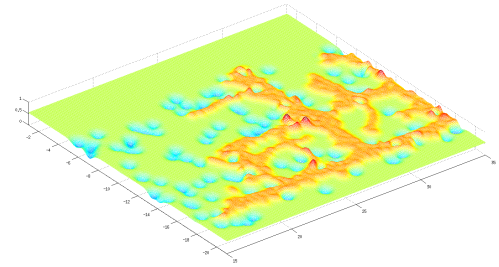

Fig. 4. Navigational map indicating the probability of each cell being free, $p = 1.0$ in *red* indicates that the area is determined to be obstacle free

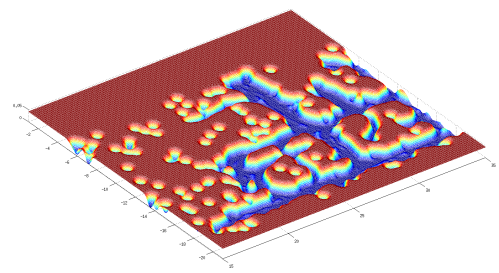

Fig. 5. Covariance of occupancy before the path is completely traversed

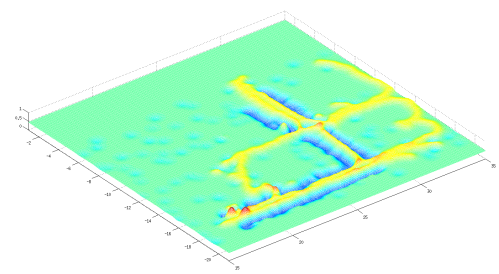

Fig. 6. Probability of occupancy after the path is completely traversed

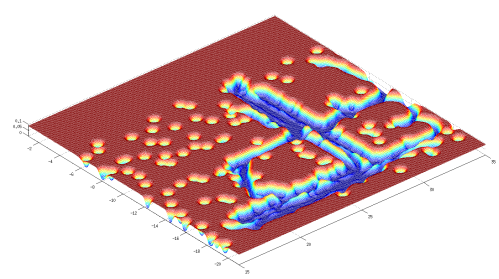

Fig. 7. Covariance of occupancy after the path is completely traversed

a execution stage of the planned trajectory is depicted in Fig. 10.

V. CONCLUSIONS AND FUTURE WORK

This paper presented a method for supplementing environment information through exploiting indirect observations of occupied space. It was shown that the proposed approach

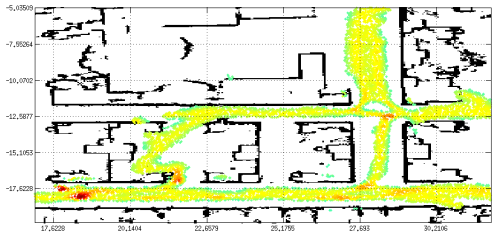

Fig. 8. Probability of occupancy superimposed on an occupancy grid map

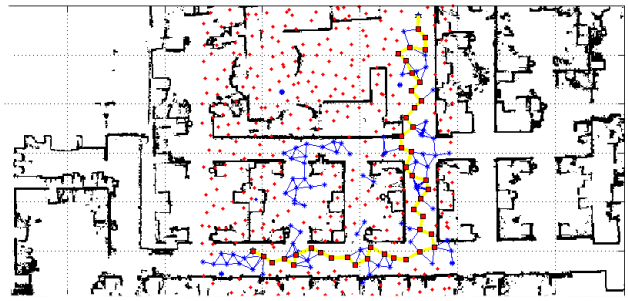

Fig. 9. Nodes of PRM in free space are noted with *blue stars*, selected nodes that are to be traversed in *red squares* and edges between selected nodes *yellow lines*

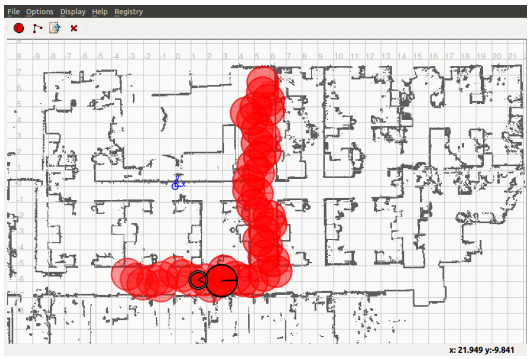

Fig. 10. A snapshot acquired during the execution of planned path by the robot, the desired path is superimposed in red, our robot is the small *red* circle and the

enables the online inclusion of novel human positional traces and/or environment information into a probabilistic framework for path planning. Furthermore, advantageous sociocontextual information, such as identifying areas that people tend to use to move through the environment, is inherently encapsulated by our method without supervised learning or labelling.

An empirical evaluation of our approach using an indoor mobile robot is presented demonstrating that a robot can efficiently and socially-appropriately reach a desired goal through exploiting the navigational map in our Bayesian statistical framework.

Whilst the results presented here within are promising, a number of limitations remain to be addressed in future work. For instance, more thorough exploitation of the sociocontextual information and investigation of using the approach in public open spaces seem viable.

References

[1] S. Thrun, M. Bennewitz, W. Burgard, A. Cremers, F. Dellaert, D. Fox, D. Hähnel, C. Rosenberg, N. Roy, J. Schulte, and D. Schulz, "MINERVA: A second generation mobile tour-guide robot," in *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, 1999.

[2] M. Montemerlo, S. Thrun, and W. Whittaker, "Conditional particle filters for simultaneous mobile robot localization and people-tracking," in *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, 2002, pp. 695 – 701.

[3] W. H. Huang, B. R. Fajen, J. R. Fink, and W. H. Warren, "Visual navigation and obstacle avoidance using a steering potential function." *Robotics and Autonomous Systems*, vol. 54, no. 4, pp. 288–299, 2006.

[4] C. Breazeal, "Toward sociable robots," *Robotics and Autonomous Systems*, vol. 42, no. 34, pp. 167 – 175, 2003.

[5] M. Luber, L. Spinello, J. Silva, and K. O. Arras, "Socially-Aware Robot Navigation: A Learning Approach," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012, pp. 1–6.

[6] T. Ikeda, Y. Chigodo, D. Rea, F. Zanlungo, M. Shiomi, and T. Kanda, "Modeling and Prediction of Pedestrian Behavior based on the Subgoal Concept," in *Robotics: Science and Systems (RSS)*, 2012, pp. 1–8.

[7] J. Müller, C. Stachniss, K. Arras, and W. Burgard, "Socially inspired motion planning for mobile robots in populated environments," in *Proc. of the International Conference on Cognitive Systems*, 2008, 1–6.

[8] X. Wang, "Learning motion patterns using hierarchical bayesian models," Ph.D. dissertation, Massachusetts Institute of Technology, 2009.

[9] S. Sehestedt, S. Kodagoda, A. Alempijevic, and G. Dissanayake, "Efficient learning of motion patterns for robots," *Proc. of the Australasian Conference on Robotics and Automation (ACRA)*, 2009.

[10] M. Bennewitz, W. Burgard, G. Cielniak, and S. Thrun, "Learning motion patterns of people for compliant robot motion," *The International Journal of Robotics Research*, vol. 24, no. 1, pp. 31–48, 2005.

[11] C. Stachniss, G. Grisetti, and W. Burgard, "Information gain-based exploration using rao-blackwellized particle filters," in *Robotics: Science and Systems*, 2005, pp. 65–72.

[12] G. Lidoris, F. Rohrmüller, D. Wollherr, and M. Buss, "The autonomous city explorer (ace) project: mobile robot navigation in highly populated urban environments," in *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, 2009, pp. 2238–2244.

[13] S. O'Callaghan, S. Singh, A. Alempijevic, and F. Ramos, "Learning navigational maps by observing human motion patterns," in *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, 2011, pp. 4333–4340.

[14] D. Hordern and N. Kirchner, "Robust and Efficient People Detection with 3-D Range Data using Shape Matching."

[15] N. Kirchner, A. Alempijevic, and A. Virgona, "Head-to-shoulder signature for person recognition," in *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2012, pp. 1226–1231.

[16] Z. Zulkarnain, S. Kodagoda, and G. Dissanayake, "Torso detection and tracking using a 2d laser range finder," in *Proc. of the Australasian Conference on Robotics and Automation (ACRA)*, 2010, pp. 1–6.

[17] S. T. OCallaghan and F. T. Ramos, "Gaussian process occupancy maps," *The International Journal of Robotics Research*, vol. 31, no. 1, pp. 42–62, 2012.

[18] C. E. Rasmussen and C. Williams, *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[19] A. Singh, F. Ramos, H. Whyte, and W. Kaiser, "Modeling and decision making in spatio-temporal processes for environmental surveillance," in *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, may 2010, pp. 5490 –5497.

[20] T. Peynot, R. Fitch, R. McAllister, and A. Alempijevic, "Resilient navigation through probabilistic modality reconfiguration," in *Proc. of the International Conference on Intelligent Autonomous Systems (IAS-12)*, 2012, pp. 1–6.

[21] N. Kirchner, A. Alempijevic, S. Caraian, R. Fitch, D. Hordern, G. Hu, G. Paul, D. Richards, S. Singh, and S. Webb, "Robotassist-a platform for human robot interaction research," *Proc. of the Australasian Conference on Robotics and Automation (ACRA*, vol. 10, pp. 1–10, 2010.