

An Extension of GHMMs for Environments With Occlusions and Automatic Goal Discovery for Person Trajectory Prediction

Ignacio Pérez-Hurtado¹, Jesús Capitán², Fernando Caballero² and Luis Merino¹

Abstract—Robots navigating in a social way should use some knowledge about common motion patterns of people in the environment. Moreover, it is known that people move intending to reach certain points of interest, and machine learning techniques have been widely used for acquiring this knowledge by observation. Learning algorithms such as Growing Hidden Markov Models (GHMMs) usually assume that points of interest are located at the end of human trajectories, but complete trajectories cannot always be observed by a mobile robot due to occlusions and people going out of sensor range. This paper extends GHMMs to deal with partial observed trajectories where people’s goals are not known a priori. A novel technique based on hypothesis testing is also used to discover the points of interest (goals) in the environment. The approach is validated by predicting people’s motion in three different datasets.

I. INTRODUCTION

Nowadays, robots working in environments populated with people are becoming commonplace. In order to show a more *social* behavior, classic navigation algorithms need to propose new objective cost functions. For instance, path planning algorithms should drive robot avoiding common people trajectories not to disturb them, or conversely, along those trajectories to search for someone [1], [2]. All these *social* navigation algorithms require the ability to predict human movements in the environment.

Human motion patterns depend typically on spatial variables: people move between doors and corridors following common trajectories, places of interest such as coffee machines are common goals, etc. Machine learning techniques can be used to infer these models from observed people trajectories.

Some approaches [3], [4] use occupancy grid maps that consider dynamic objects and represent the probability of people appearance and disappearance for each cell. In [3], a multi-layer framework for social navigation is introduced. Each layer has a purpose: one is used to determine the accessible area of the environment; another one is used to represent rates of people appearing; and the last one represents a place-dependent reliability of the person detector on board the robot. Authors in [5] present a system of multi-probability

grids for social mapping. Motion probability grids represent the potential motion of a human moving to a particular target grid cell.

Agent-based and velocity-space reasoning are also typically used as in [6], where each pedestrian is modeled as an agent using statistical inference techniques. Thus, the robot is able to learn individual motion parameters for every agent in the scene, providing a solution for a collision-free robot navigation. In [7], clustering techniques are applied over pedestrian motion data to extract information about the use of the space and people’s typical behavior. Then, this is used by a robot to predict likely behaviors in certain places and offer services to idle people at a shopping center. In [8], the authors describe a behavior cognition model to represent the spatial effects in the relations between people and also between people and the environment.

Motion patterns and people intentions are affected by points of interest in the environment, and thus, the problem of estimating such points is also considered in the literature [9]. These points may not have discriminative appearance and shape, but they affect the behavior of people in the scene, attracting them (e.g., vending machines) or repelling them (e.g., grass lawns).

Hidden Markov Models (HMMs) were one of the first machine learning techniques used to estimate typical motion patterns [10]. In particular, we focus in this paper on Growing Hidden Markov Models (GHMMs) [11], which can learn the spatial structure of trajectories in a specific environment by building an Instantaneous Topological Map (ITM) that can be viewed as a dynamic occupancy grid map. First, a learning phase trains an HMM that is iteratively fed with observed trajectories; then, a prediction phase takes new trajectories and determines the probability distribution of next positions in a time horizon, as well as the probability distribution of future goals. GHMMs can work in an online fashion, training the model at the same time that predictions are computed. Moreover, not only the parameters of the HMM are trained, but also the ITM and HMM structures evolve with new observations.

One of the main issues with current GHMM implementations is the need for complete person trajectories (including actual starting and ending/goal points) in the training phase. The GHMM training algorithm does not apply any special reasoning about where the points of interest are located, since it is assumed that goals are the final points of the trajectories. Tracking complete trajectories can be achieved

¹I. Pérez-Hurtado and L. Merino are with the Pablo de Olavide University, Seville, Spain {iperde,lmercab}@upo.es.

²J. Capitán and F. Caballero are with the University of Seville, Seville, Spain {jcapitan,fcaballero}@us.es.

This work is partially funded by the EC-FP7 under grant agreement no. 611153 (TERESA) and the project PAIS-MultiRobot, funded by the Junta de Andalucía (TIC-7390). I. Pérez-Hurtado is also supported by the Postdoctoral Junior Grant 2013 co-funded by the Spanish Ministry of Economy and Competitiveness and the Pablo de Olavide University.

when the whole environment is observable at once (e.g., using a zenithal camera [12]), but tracking people based on sensors on board a mobile robot could lead to a poor learning due to occlusions that interrupt the observed trajectories. In this paper, we propose an extension of GHMMs designed to be used with on-board robot sensors. A first contribution is a novel learning phase that uses partial trajectories where it cannot be determined whether individuals have reached their final goals or not (they may go out of the sensor range or get occluded). We also contribute with a new hypothesis testing method to estimate potential points of interest in the environment (goal points of trajectories). Finally, a more accurate modeling of the sensors has been included, since the observation function associated with each topological node is dynamically updated during the learning phase.

Experimental results are provided for two public datasets with zenithal cameras, and a dataset recorded using a robot in an indoor office-like environment. A couple of metrics have been used to evaluate the quality of the prediction of people’s motion with our model.

The remainder of the paper is as follows: Section II summarizes the concepts related to GHMMs; Section III describes a new improved version of the original ITM algorithm; Section IV presents the hypothesis testing approach for discovering goals of trajectories; Section V explains how to train the HMM and make future predictions; Section VI provides experimental results; and Section VII discusses conclusions and future work.

II. GROWING HIDDEN MARKOV MODELS

In a GHMM, there is a discrete representation of the space, which is divided into regions. Transitions are only allowed between neighboring regions. The learning process consists of estimating the best space discretization, and identifying neighboring regions and transition probabilities from observed data. First, a topological map is built with the ITM algorithm [13]; then, an HMM is built from the ITM, and its transition (and prior) probabilities are trained with the incremental Baum-Welch technique [14]. Moreover, in GHMMs (contrary to HMMs) the number of discrete states and transitions are not constant, i.e, the structure of the model changes as more input observations are processed.

The general steps of the complete algorithm for learning a GHMM are summarized in the following subsections. Further details about how to train a GHMM and make predictions with it can be seen in [11].

A. Updating the topological map

A topological map is a discrete representation of the space in the form of a graph where nodes represent regions (each node has associated the centroid of the region) and edges connect nodes related to adjacent regions. The ITM algorithm [13] updates iteratively a topological map given the observed data. Each observation O_t consists of a 2D position O^p of a detected person at a given time step t ; and a trajectory $O_{1:T}$ is a sequence of observations of the same person along T time

steps. As new observations arrive, nodes and edges may be created (or moved) or deleted in the topological map.

B. Updating the HMM structure

A GHMM consists of an ITM and an underlying HMM, where the states and transitions of the HMM correspond to the nodes and edges of the ITM (each edge corresponds to two transitions, since the ITM graph is undirected but the HMM graph is directed). During the ITM update, nodes and edges may be added or deleted, and those changes in the topological map must be translated to the HMM structure. Thus, for each new node in the ITM, a new state is added in the HMM with a default prior probability and a self-transition probability; for each removed node, the corresponding state is removed. Moreover, for each new edge in the ITM, two transitions are added in the HMM with default probabilities; for each removed edge, the corresponding transitions are removed. Then, all priors and transition probabilities are normalized.

It is assumed that people move intending to reach a particular goal. Thus, each node i in the ITM (or HMM state) contains 4-dimensional spatial information (x_i, y_i, g_i^x, g_i^y) , where (x_i, y_i) is the centroid c_i of the current region and (g_i^x, g_i^y) is the intended goal related to that node. Also, for every node/state i , an associated Gaussian distribution $G(c_i, \Sigma)$ is stored representing the likelihood of observations for that state $P(O^p|i)$. The covariance Σ is considered to be fixed and the same for all nodes.

C. Updating the HMM probabilities

In a GHMM, the prior and transition probabilities of the states in the HMM are learnt from data (the observed people trajectories). This step to update the unknown parameters of the HMM can take place once per complete observed trajectory $O_{1:T}$, after all the discrete observations O_t contained in the trajectory have been processed to update the ITM and the HMM structures. An incremental version of the Baum-Welch algorithm [14] is used to re-estimate the probabilities of the HMM. Moreover, as the observations do not have information about the goals of the people, before this update, each observation O_t of a complete observed trajectory $O_{1:T}$ is extended by assigning as goal coordinates the last position of the trajectory O_T .

D. Predicting motion

Once the underlying HMM is learnt, that model can be used to filter and predict future positions of people by using inference. Given an initial estimation of the person position, after every new observation of the person position, the current belief on the person position and his/her goal are re-estimated by applying Bayes recursion. Predictions can be performed by propagating this estimation a number of time steps ahead into the future.

III. IMPROVED INSTANTANEOUS TOPOLOGICAL MAP

One of the main assumptions in the original GHMM is that the goals of people are given by the last positions of

the observed trajectories, and hence, complete trajectories between origins and goals are required. Here we modify the ITM to include information about the flux of people, which will allow us to automatically discover goals (points of interest) considering only partial (incomplete) observed trajectories. This novel procedure to build the ITM is described in Algorithm 1.

At each node i , a tuple $(i_n, i_{in}, i_{out}, i_s)$ is stored. i_n represents the number of different people that are observed in that node, while i_{in} is the number of people that appear at that node and i_{out} the number of people that disappear. We say that a person appears at the node where it is registered the first observation for that person. On the other hand, we say that a person disappears at the node where the algorithm gets the last observation for that person. The last observation is determined if there are no more observations of that person after a t_{out} period. Finally, i_s is the mean time that the people detected stand at that particular node.

In addition, instead of using a fixed Gaussian as likelihood function $P(O^p|i)$, we extend the ITM to store and update a specific covariance matrix for each node, so the observation model can adapt to the characteristics of the different parts of the scenario. The centroids are also updated in a different manner as in the original ITM [13], and each centroid is computed as the mean of its associated observations.

A. Gaussian distribution adaptation

Each node i stores a bivariate Gaussian $G(c_i, \Sigma_i)$ as its observation likelihood $P(O^p|i)$, with:

$$\Sigma_i = \begin{bmatrix} \sigma_x^2 & \rho\sigma_x\sigma_y \\ \rho\sigma_x\sigma_y & \sigma_y^2 \end{bmatrix}_i$$

For each new observed position $O^p = (x, y)$, Algorithm 1 updates the parameters of the Gaussian distribution as follows:

$$\begin{aligned} c'_i &= c_i = (c_x, c_y)_i \\ c_i &= (c_x + \frac{x - c_x}{i_n}, c_y + \frac{y - c_y}{i_n}) \\ \rho &= \rho + \frac{(x - c_x)(y - c_y) - \rho}{i_n} \\ s_x &= s_x + (x - c'_x)(x - c_x) \\ s_y &= s_y + (y - c'_y)(y - c_y) \\ \sigma_x &= \sqrt{\frac{s_x}{i_n - 1}} \\ \sigma_y &= \sqrt{\frac{s_y}{i_n - 1}} \end{aligned}$$

where s_x and s_y are initialized to 0 at the first iteration. Basically, the Gaussian of each node comes from computing the average and standard deviation of all observed positions at that node.

Algorithm 1 Improved ITM Algorithm

Require: Person identifier id , observed position O^p , insertion threshold τ , timeout t_{out} .

- 1: **Matching:** Find the nearest node j and the second-nearest node k with respect to the observed position O^p in terms of Euclidean distance.
 - 2: **Edge adaptation:** (i) Create an edge connecting j and k if it does not already exist. (ii) For each node m in the neighborhood of j (neighbors are those 1-hop-connected), check if the centroid of k lies inside the Thales sphere through the centroids of j and m . If that is the case, remove the edge connecting j and m . When deleting an edge, check m for emanating edges; if there are none, remove that node as well.
 - 3: **Node adaptation:** If O^p lies outside the Thales sphere through the centroids of j and k , and outside a sphere around the centroid of j with a given radius τ , create a new node i with $i_n = 0$, $i_{in} = 0$, $i_{out} = 0$, $i_s = 0$. Connect nodes i and j . If the centroids of j and k are closer than $0.5*\tau$, remove k . Select a node r to be adapted as $r = i$ if i was created, and $r = j$ otherwise.
 - 4: **Gaussian distribution adaptation:** Adapt the centroid and covariance matrix associated with node r according to the method explained in Section III-A.
 - 5: **In/out people adaptation:** $r_n = r_n + 1$. If it is the first time observing the person id , then $r_{in} = r_{in} + 1$, and set to 0 a time counter associated with id . When the time counter is greater than t_{out} , $r_{out} = r_{out} + 1$ and remove the time counter.
 - 6: **Standing time adaptation:** When the person abandons the node, update $r_s = r_s + \frac{\Delta t - r_s}{r_n}$, with Δt the total time the person was observed at the same node.
-

IV. DISCOVERING GOALS

When there are partial observed trajectories, people's starting points and goals are not known a priori, so a procedure to discover them is needed. It cannot be assumed that a person who disappears at a node has arrived to his/her goal. He/she could be out of the range of the sensors or occluded. In the same way, a person that appears at a node is not necessarily a new person, because he or she could just be back after being out of range or occluded.

If people are detected and tracked with the sensors on board a mobile robot, it could be assumed that the number of people appearing/disappearing at actual entry/goal points (e.g., doors) should be significantly higher in the long term than the number of people appearing/disappearing at false entry/goal points produced by occlusions or losses of tracking. Moreover, goals are not only areas where people exit the environment; goals can also be considered as points of interest where people spend more time than normal, for instance, standing in front of a coffee machine. Therefore, it can be assumed that the time spent in this type of goals should be significantly higher in the long term than the average time in other nodes.

In order to determine the significance level of each node to be considered a goal point, three hypothesis tests (t-tests) are applied. Nodes with a p-value less than a threshold parameter in one or more of the hypothesis tests are considered goal points (entry points are also considered as goals, since they may be exit points for other people in the future). The p-values for all the nodes are re-calculated after each iteration of the ITM algorithm. Thus, existent goals could be also removed.

A. Discovering entry/exit points

For each node i , the mean of the relative frequency of people appearing $\mu_{in} = \frac{i_{in}}{i_n}$ is computed. The associated standard deviation is computed too:

$$\sigma_{in} = \sqrt{\frac{i_{in}(1 - \mu_{in})^2 + (i_n - i_{in})\mu_{in}^2}{i_n - 1}}$$

Then, the next hypothesis test is formulated to check whether the node is an entry point:

- H_0 : μ_{in} is less than or equal to p_{in}
- H_1 : μ_{in} is greater than p_{in}

The value p_{in} is a parameter of the algorithm. The idea is that the relative frequency of people appearing at an entry node should be clearly higher than that of a node where people pass by (with $p_{in} \approx 0.5$). The hypothesis test to determine whether a node is an exit point is analogous to the one used for entry points, but using i_{out} instead of i_{in} .

B. Discovering standing points

During the tracking process, each node i stores and updates its average standing time i_s , i.e, the average time people stand in the area corresponding to that node. Its associated standard deviation is also calculated. Thus, the next hypothesis test can be formulated to check whether a node i is a standing point:

- H_0 : i_s is less than or equal to T_s seconds.
- H_1 : i_s is greater than T_s seconds.

The value of T_s seconds is a parameter of the algorithm.

V. HMM LEARNING AND PREDICTION

Once the ITM and the goal points have been updated, the underlying HMM can also be updated accordingly. The (hidden) states in the HMM are all the possible combinations (n, p) , where n is a node of the ITM and p is a discovered goal. Therefore, the number of states in the HMM is $N * G$, where N is the number of nodes in the topological map and G is the number of discovered goals.

The transition probability to go from a state $S = (n, p)$ to a state $S' = (m, q)$ is zero if $S \neq S'$ and there is no edge connecting n and m in the ITM. Therefore, the number of possible non-zero transitions in the HMM are $(2E + N)G^2$, where E is the number of edges in the ITM.

A. Updating the HMM structure

The HMM structure is updated following the next rules:

- For each new node n in the ITM, create all possible states (n, p) , where p is contained in the list of discovered goals. Assign a default initial prior probability to each created state and initiate all the associated transition probabilities to a default value if there exists the corresponding edge in the ITM.
 - For each removed node n in the ITM, remove all the states (n, p) and assign a zero value to all the associated prior and transition probabilities.
 - For each new goal p , create all possible states (n, p) , where n is one of the nodes in the ITM. Assign a default initial prior probability to each new state and initiate all the associated transition probabilities with a default initial value if there exists the corresponding edge in the ITM.
 - For each removed goal p , remove all the states (n, p) and assign a zero value to all the associated prior and transition probabilities.
 - For each new edge (n, m) in the ITM, assign a default initial value to all possible associated transition probabilities.
 - For each removed edge (n, m) in the ITM, assign a zero value to all the associated transition probabilities.
- All the probabilities will be normalized in the next step.

B. Updating the HMM probabilities

After each partial observed trajectory, the HMM state prior and transition probabilities can be re-estimated following the incremental Baum-Welch algorithm, as it is done in [11]. However, we consider observations with information about the position and velocity of the person $O_t = (O^p, O^v)$. The observation probability for a given state $P(O_t = (x, y, v_x, v_y) | S_t = (n, p))$, which is needed to calculate the forward and backward probabilities in the Baum-Welch algorithm, can be defined as:

$$P((x, y, v_x, v_y) | (n, p)) = P(v_x, v_y | x, y, n, p)P(x, y | n, p)$$

The second probability does not depend on the goal p , and is given by the Gaussian distribution corresponding to node n in the ITM (see Section III-A):

$$P(x, y | n, p) = P(x, y | n) = G(c_n, \Sigma_n)$$

For the first probability we assume that the velocity vector of the person should point to the goal of the trajectory. In this sense, a Gaussian bivariate distribution $G(\mu_v, \Sigma_v)$ is used, where:

$$\mu_v = \left(\frac{p_x - x}{\sqrt{(p_x - x)^2 + (p_y - y)^2}}, \frac{p_y - y}{\sqrt{(p_x - x)^2 + (p_y - y)^2}} \right)$$

$$\Sigma_v = \begin{bmatrix} \sigma_x'^2 & 0 \\ 0 & \sigma_y'^2 \end{bmatrix}$$

Note that (p_x, p_y) is the position of the goal point p , while σ_x' and σ_y' are estimated depending on the distance to the goal.

C. Prediction of people motion and trajectory goals

In the HMM, a belief over the state at each time step is maintained. After a partial observed trajectory, that belief is re-estimated (η is a normalizing constant):

$$P(S_t|O_{1:t}) = \frac{1}{\eta} P(O_t|S_t) \sum_{S_{t-1}} P(S_t|S_{t-1})P(S_{t-1}|O_{1:t-1})$$

The belief over the current area (node) where the person is located can also be defined:

$$P(n_t|O_{1:t}) = \frac{1}{\eta} \sum_p P(S_t = (n, p)|O_{1:t})$$

Similarly, the belief over the distribution of possible goals would be:

$$P(p_t|O_{1:t}) = \frac{1}{\eta} \sum_n P(S_t = (n, p)|O_{1:t})$$

Then, predictions over the future positions of the people in the scenario can be performed by propagating the estimation H time steps ahead into the future:

$$P(S_{t+H}|O_{1:t}) = \sum_{S_{t+H-1}} P(S_{t+H}|S_{t+H-1})P(S_{t+H-1}|O_{1:t})$$

and computing again the beliefs over the current area and possible goal of the observed trajectory.

VI. EXPERIMENTAL RESULTS

We implemented our approach in C++ under the Robot Operating System (ROS) framework, and we run experiments with three different datasets: two of them public (Edinburgh Informatics Forum Pedestrian Dataset [12] and the BIWI Walking Pedestrian Dataset [15]); and another one gathered by the TERESA robot [16] at the Pablo de Olavide University.

A. Edinburgh Informatics Forum Pedestrian Dataset

This dataset consists of people walking through the Informatics Forum, the main building of the School of Informatics at the University of Edinburgh, recorded at 9 fps. The dataset is composed by the images and the people's positions with unique ids, allowing easy person tracking and goal detection for trajectories.

In order to train our models, we used 2,000 trajectories obtaining a topological map of 375 nodes, 962 edges and 9 discovered goals, which is shown at Fig. 1. The associated HMM contains 3,375 states and 159,219 non-zero transition probabilities.

B. BIWI Walking Pedestrian Dataset

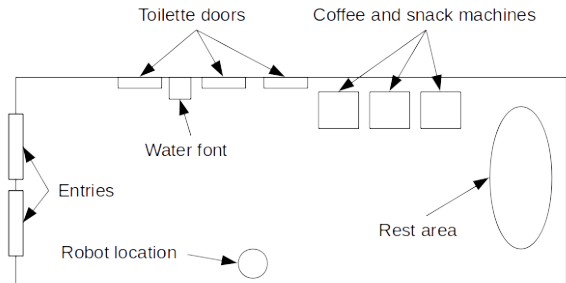
This dataset was recorded from the top of the ETH main building (Zurich) at 2.5 frames per second. As before, the dataset provides people's positions with unique ids, allowing easy trajectory computation. We used 160 trajectories to train an ITM of 223 nodes, 517 edges and 5 discovered goals. The HMM contains 1,115 states and 26,965 non-zero transitions.



Fig. 1. Topological map (blue) and discovered goals (red) for the Edinburgh dataset.



(a) Coffee machine area



(b) Coffee machine area scheme

Fig. 2. Coffee machine area at UPO.

C. Pablo de Olavide University

The coffee machine area in the building 45 of the Pablo de Olavide University (UPO) (Fig. 2a) is an area of 4.30×11.80 meters with an entry point from two corridors and several points of interest: three coffee and snack machines, three toilette doors, a water font and a rest area with chairs and bar-style tables.

In this experiment, a robot [16] was static in front of the toilette doors as shown in Fig. 2b. Only an onboard laser-

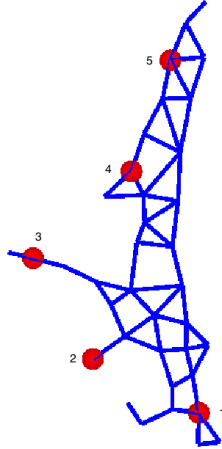


Fig. 3. Topological map (blue) and discovered goals (red) for the coffee machine area at UPO.

scanner was used for person detection and tracking based on [17] and a Kalman filtering for temporal tracking and velocity estimation. A total of 20 trajectories were recorded in a dataset and used to train our models, generating an ITM with 37 nodes, 151 edges and 5 discovered goals (Fig. 3). The discovered goals correspond to (1) the entry point, (2) the water font, (3) the man’s toilette door, (4) the coffee/snack machines and, (5) the rest area. The related HMM contains 185 states and 2,725 non-zero transition probabilities.

D. Validation results

The objective of this section is to provide a quantitative criteria to assess the accuracy of the predictions based on the models learnt previously. The two first datasets were used for this. They incorporate the final goal of every person, which was used as ground-truth to evaluate predictions. Nonetheless, this information was not used to build the ITM nor the HMM.

A subset of the trajectories were not used in the training phase and are used here to evaluate the prediction phase. Particularly, 200 trajectories for Edinburgh and 159 for BIWI. Two are the main factors we can evaluate with these test datasets: (1) how accurately the method discovers the position of the different goals in the models and (2) the accuracy of the goal prediction during a person trajectory.

Regarding the automatic goal discovery rate, in the BIWI dataset, the 58% of the total goals were automatically detected in the training phase. Moreover, the 64% of the total goals were detected for the Edinburgh dataset. Even though the computed models fail for unusual patterns, they are able to represent the main and more repeated trajectories. Larger datasets would probably achieve a better goal discovery rate.

Regarding trajectories with ground-truth (actual goals), we want to evaluate how accurately our approach predicts the person’s goal based on his/her trajectory. For this purpose, people’s trajectories were divided into segments of 25%, 50%, 75% and 100% of their length. The predictions of our

algorithm were evaluated at these points and compared to the ground-truth. Two metrics were computed to assess this prediction phase:

- **Metric 1:** $\sum_{g \in G} P(g|O_{1:t})k_g$
- **Metric 2:** $\sum_{g \in G} P(g|O_{1:t})d_g$

where:

- G is the set of possible goals.
- $P(g|O_{1:t})$ is the probability of goal g to be the goal of the trajectory $O_{1:T}$ based on the belief at time t .
- k_g is 1 if g is the actual goal of the trajectory $O_{1:T}$, 0 otherwise.
- d_g is the Euclidean distance of the centroid of g to the centroid of the goal of $O_{1:T}$.

Metric 1 evaluates how accurately the approach predicts the correct goal (regarding the ground-truth), providing the probability of sampling the right goal conditioned to all the previous observations. The closer to 1 the result of Metric 1 is, the better. Metric 2 tries to assess the accuracy of the position of the predicted goals, that means, if the predicted goal is close or far from the actual goal. We use the Euclidean distance in meters for Metric 2, and the smaller the value, the better.

Tables I and II show the outcome of these metrics for our experiments. The tables depict average values and standard deviations of each metric after running all the test trajectories. It can be seen how the prediction is clearly improved with the length of the trajectories for the Edinburgh dataset, while the prediction in the BIWI dataset is very good for all cases. It can also be seen how the standard deviations of the metrics also decrease with the length of the segment, as expected.

In addition, a hypothesis test (t-test) was run for Metric 1, contrasting the obtained average values with the uniform distribution:

- H_0 : The average value of Metric 1 is less than or equal to 1/9 in the case of the Edinburgh dataset or 1/5 in the case of the BIWI dataset.
- H_1 : The average value of Metric 1 is greater than 1/9 in the case of the Edinburgh dataset or 1/5 in the case of the BIWI dataset.

The p-values are shown in the last column of Tables I and II. It can be seen how the value is very small (insignificant) for all the cases.

Trajectory	Metric 1		Metric 2		p-value
	mean	σ	mean	σ	
25%	0.4774	0.4259	3.4465	3.6886	$< 2.2 * 10^{-16}$
50%	0.5857	0.4264	2.3300	2.9505	$< 2.2 * 10^{-16}$
75%	0.8381	0.3223	0.8850	2.2087	$< 2.2 * 10^{-16}$
100%	0.9937	0.0511	0.0647	0.5335	$< 2.2 * 10^{-16}$

TABLE I
EDINBURGH DATASET EXPERIMENTS.

VII. CONCLUSIONS

This paper has presented a system for person trajectory prediction. The system extends GHMMs to deal with partial

Trajectory	Metric 1		Metric 2		p-value
	mean	σ	mean	σ	
25%	0.8983	0.2351	1.1276	2.8390	$< 2.2 * 10^{-16}$
50%	0.9247	0.2112	0.7643	2.6173	$< 2.2 * 10^{-16}$
75%	0.9425	0.1898	0.7641	3.0416	$< 2.2 * 10^{-16}$
100%	0.9802	0.1095	0.3381	1.8972	$< 2.2 * 10^{-16}$

TABLE II
BIWI DATASET EXPERIMENTS.

observed trajectories when training the model and making predictions. Indeed, partial trajectories are usual when only local sensing on board a robot is used for person detection and tracking. Thus, this approach makes GHMMs more robust against occlusions and miss-detections. Moreover, partial trajectories may not include final people's goals, which are automatically detected by the system. Those goal points are typically points of interest in the environment (vending machines, doors, etc.), and are considered within the model, since they affect the people motion.

The method has been benchmarked using different public datasets, showing the results that a good number of goals are automatically detected. The results also showed that goal prediction based on person trajectory is consistent, and more accurate as the partial trajectories get closer to the complete one. Besides, the method has been evaluated qualitatively using data from a robot in an indoor scenario.

As future work, the model will be integrated with the person tracker itself, so it can be used in the prediction phase, leading to more robust trackers. Furthermore, this model for people motion will be the base for human-aware path planning and task planning under uncertainties in missions involving human and robots.

REFERENCES

- [1] G. D. Tipaldi and K. O. Arras, "Planning problems for social robots," in *Proc. International Conference on Automated Planning and Scheduling (ICAPS'11)*, Freiburg, Germany, 2011.
- [2] L. Merino, J. Ballesteros, N. Pérez-Higueras, R. Ramón-Vigo, J. Pérez-Lara, and F. Caballero, "Robust person guidance by using online POMDPs," in *ROBOT2013: First Iberian Robotics Conference*, ser. Advances in Intelligent Systems and Computing, M. A. Armada, A. Sanfeliu, and M. Ferre, Eds. Springer International Publishing, 2014, vol. 253, pp. 289–303.
- [3] M. Luber, G. Tipaldi, and K. Arras, "Place-dependent people tracking," *The International Journal of Robotics Research*, vol. 30, no. 3, pp. 280–293, 2011.
- [4] D. Meyer-Delius, M. Beinhofer, and W. Burgard, "Occupancy grid models for robot mapping in changing environments," in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, Toronto, Canada, 2012.
- [5] S. Thompson, T. Horiuchi, and S. Kagami, "A probabilistic model of human motion and navigation intent for mobile robot path planning," in *Proceedings of the International Conference on Automation, Robotics and Applications (ICARA)*, G. S. Gupta and S. C. Mukhopadhyay, Eds. IEEE, 2009, pp. 663–668.
- [6] S. Kim, S. Guy, W. Liu, D. Wilkie, R. Lau, M. Lin, and D. Manocha, "Brvo: Predicting pedestrian trajectories using velocity-space reasoning," *The International Journal of Robotics Research*, vol. 34, no. 2, pp. 201–217, 2015.
- [7] T. Kanda, D. Glas, M. Shiomi, and N. Hagita, "Abstracting people's trajectories for social robots to proactively approach customers," *Robotics, IEEE Transactions on*, vol. 25, no. 6, pp. 1382–1396, Dec 2009.
- [8] S.-Y. Chung and H.-P. Huang, "A mobile robot that understands pedestrian spatial behaviors," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*. IEEE, 2010, pp. 5861–5866.
- [9] D. Xie, S. Todorovic, and S. Zhu, "Inferring "dark matter" and "dark energy" from videos," in *IEEE International Conference on Computer Vision (ICCV)*, 2013, pp. 2224–2231.
- [10] M. Bennewitz, W. Burgard, G. Cielniak, and S. Thrun, "Learning motion patterns of people for compliant robot motion," *The International Journal of Robotic Research*, vol. 24, no. 1, pp. 31–48, 2005.
- [11] D. Vasquez, T. Fraichard, and C. Laugier, "Growing hidden markov models: An incremental tool for learning and predicting human and vehicle motion," *The International Journal of Robotics Research*, vol. 28, no. 11-12, pp. 1486–1506, 2009.
- [12] Edinburgh informatics forum pedestrian dataset. [Online]. Available: <http://homepages.inf.ed.ac.uk/rbf/FORUMTRACKING/>
- [13] J. Jocksuch and H. Ritter, "An instantaneous topological map for correlated stimuli," in *Proceedings of the International Joint Conference on Neural Networks*, vol. 1, Washington, DC, 1999, pp. 529–534.
- [14] R. Neal and G. Hinton, "A view of the em algorithm that justifies incremental, sparse, and other variants," in *Learning in Graphical Models*, M. Jordan, Ed., 1998.
- [15] Biwi walking pedestrian dataset. [Online]. Available: <http://www.vision.ee.ethz.ch/datasets/index.en.html>
- [16] K. Shiarlis, J. Messias, M. van Someren, S. Whiteson, J. Kim, J. Vroon, G. Englebienne, K. Truong, V. Evers, N. Perez-Higueras, I. Perez-Hurtado, R. Ramon-Vigo, F. Caballero, L. Merino, J. Shen, S. Petridis, M. Pantic, L. Hedman, M. Scherlund, R. Koster, and H. Michel, "Teresa: A socially intelligent semi-autonomous telepresence system," in *Workshop on Machine Learning for Social Robotics at ICRA-2015 in Seattle*, 2015.
- [17] K. O. Arras, O. M. Mozos, and W. Burgard, "Using boosted features for the detection of people in 2D range data," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Roma, Italy, April 2007, pp. 3402–3407.