Better Motion Prediction for People-tracking

Allison Bruce
Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213
Email: abruce@ri.cmu.edu

Geoffrey Gordon
Center for Automated Learning and Discovery
Carnegie Mellon University
Email: ggordon@cs.cmu.edu

***Abstract*— An important building block for intelligent mobile robots is the ability to track people moving around in the environment. Algorithms for person-tracking often incorporate motion models, which can improve tracking accuracy by predicting how people will move. More accurate motion models produce better tracking because they allow us to average together multiple predictions of the person's location rather than depending entirely on the most recent observation. Many implemented systems, however, use simple conservative motion models such as Brownian motion (in which the person's direction of motion is independent on each time step). We present an improved motion model based on the intuition that people tend to follow efficient trajectories through their environments rather than random paths. Our motion model learns common destinations within the environment by clustering training examples of actual trajectories, then uses a path planner to predict how a person would move along routes from his or her present location to these destinations. We have integrated this motion model into a particle-filter-based person-tracker, and we demonstrate experimentally that our new motion model performs significantly better than simpler models, especially in situations in which there are extended periods of occlusion during tracking.**

I. Introduction

Accurately tracking moving people is of critical importance to robots. Knowing a person's current position or being able to anticipate a person's future position is useful for navigation in populated areas. Many tracking methods perform well when it is possible to observe the person being tracked at each time step. However, in realistic environments, people's trajectories are often partially occluded. Long periods of occlusion can lead to a person becoming lost by the tracker and falsely identified as a different person when they return to view. We assert that in many situations, this problem can be avoided by using an improved motion model which gives the tracker a better idea of where to look for the person when they return to view.

When people move through familiar environments, they do not wander according to brownian motion or always continue to move in the direction that they are currently moving. Yet many trackers naively make these assumptions in their motion models. Typically, people in public buildings walk between a finite number of points of interest (e.g., doors, corridors), often following set paths that are determined by a combination of practicality and unwritten social rules. Their motion is goal-oriented, and if we use that information in our motion model, we are more likely to successfully track a person despite periods of occlusion.

Information about the location of goals can be obtained by clustering a set of recorded trajectories. We take advantage of the goal-directed nature of this motion and use the goals as a concise representation of the trajectories. Motion updates are obtained planning a path from a person's location to the set of goals.

The method we propose consists of two steps. In the training phase, given a set of previously collected trajectories, the goal locations are fit using an algorithm that approximately maximizes their likelihood. In the tracking phase, a planner plans paths from the person's last observed position to these goals, and the resulting paths are used in the motion model of our tracker. We will explain the details of this approach, suggest when the assumptions on which it relies may and may not hold, and present results which show improved tracking performance for an experiment in which a physical robot tracks real people using a particle filter.

II. Related Work

Most work using Bayes filters for people tracking assumes a Brownian motion model [1], or a first order motion model such as would be used with a Kalman filter [2]. Both of these simple models have limitations. The Brownian model is extremely conservative and does not attempt to model the dynamics of human motion. Because all of the motion is represented as dispersion, the hypotheses become equally spread out over a broad area when there are no observations. This is often a poor estimate of the actual distribution over a person's possible position because people do not move randomly. Additionally, the expected distance travelled by a particle under the Brownian motion model increases as $\sqrt{n}$, where n is the number of timesteps. This means that the longer a moving person is unobserved, the less likely the displacement predicted by Brownian motion is to be accurate. A first order model of a person's dynamics is less conservative, and therefore often more accurate. But it assumes that a person will continue to move in the direction that they were last observed moving in. People often turn corners and avoid obstacles. There is no way to represent those kinds of actions with a first order model.

More sophisticated models of human motion for tracking have been proposed. A piecewise-linear gaussian mixture has been used as the motion model for a person-tracking task using

Wavelan signal strength as the sensor [3]. The model uses a gaussian to represent the probability of each possible action in a cell (go forward, go backward, go left, go right, stop) of a coarse grid over the map of the environment. While this approach works for Wavelan-based tracking because the high error in localization necessitates a coarse discretization of the space, it would be difficult to collect enough data to learn an accurate model at the fine discretization needed to be useful for tracking with a more accurate sensor. Also, this motion model treats future actions as only dependent on the current location, rather than viewing both as part of a path that is moving a person towards a chosen destination many steps in the future.

Another learned motion model is described by Liao et al [4]. They track people using a particle filter with a motion model that is constrained to move only along the voronoi graph of the environment. The parameters of this motion model are trained using the EM algorithm. The voronoi constraint works well in this case because their goal is to recover high-level motion behavior (e.g., which rooms a person has visited) using sparse and noisy sensors rather than to recover a person's location with a much metric accuracy. One thing that differentiates both this approach and the one described above from ours is that the movements represented in these motion models are fixed and constrained to a coarse set of possible directions. By using high level goals and a path planner in the motion model, the major axes of motion are dependent on a person's current location. This allows a greater range of possible motion directions, which makes the motion model more likely to make accurate proposals from a variety of locations.

Learning a model of common paths through through a space was first proposed for service robots by (Bennewitz, 2002). The model used for paths in that paper was a closely-spaced sequence of waypoints with Gaussian errors. This model was used to classify which group of trajectories a tracked trajectory belonged to, but it was not used to improve the performance of the underlying tracker. Additionally, the sequence-of-waypoints model is less expressive than the model we propose here: taking into account the goal-directed nature of human motion means that each cluster needs far fewer parameters while predicting motion accurately over a wider area.

While there is a considerable amount of work on appearance-based people-tracking, we will not discuss it because the issues involved are significantly different from the range-sensor based tracking that is used for our application.

III. Tracking using Bayes Filters

A. Bayes Filters

The use of Bayes filters (Kalman filters, particle filters, etc.) for tracking is common. These approaches combine prior information about the state history with observations to come to a new estimate of the state. The Markov assumption allows the current state to represent the entire history, reducing computational cost.

$$p(x_t|z^t) = \mu_t p(z_t|x_t) \int p(x_t|x_{t-1}) p(x_{t-1}|z^{t-1}) dx_{t-1}$$

The normalizing constant, μ, is given by

$$\mu = 1/p(z_t|z^{t-1}) = \int p(z_t|x_t) p(x_t|z^{t-1}) dx_t$$

and is dependent on the likelihood function, $p(z_t|x_t)$. The normalization constant is a good measure of accuracy for a bayes filter because it is proportional to the probability of the next observation given the last state.

The two modeling choices that determine the characteristics of the filter are the choice of the measurement model, $p(z_t|x_t)$, and the motion model, $p(x_t|x_{t-1})$.

While some filters control the complexity of the posterior distribution by making simplifying assumptions about the form of the motion and observation models, particle filters instead approximately represent arbitrarily complex posterior distributions with sets of state samples, or particles [5]. Particle filters are especially useful for our application because they can be used to represent distributions that cannot be expressed in closed form.

B. Tracking

The primary contribution of our approach is the proposal of a complex motion model that is trained on prior examples of people's movements in an area. Our approach is based on the assumption that people's movements through a space can be represented at a high level as progress towards one of a finite set of goal locations. Paths from a person's current location to these goals (such as can be obtained by a planner) give use information about that person's possible future position. Each sample, or particle, in our filter represents a hypothesis about the person's current state. The state is made up of the person's location and which goal they are moving towards. We assume that a particle's goal is static and cannot change as long as that hypothesis exists.

The likelihood that a person will approach a particular goal location can be easily estimated from the data during the training phase. This prior distribution over possible goals is represented by the percentage of particles that have each goal in their state when the filter is initialized.

The motion update propagates a hypothesis along the path returned by the planner to its goal. First, the line segment of the plan that is closest to the location of the hypothesis, l_{nearest}, is determined, and the direction of that line segment is found. The hypothesis is then displaced in that direction by a distance randomly sampled from a gaussian, d, and then a small amount of gaussian noise is added to that location, n_x, n_y. In our implementation, σ_d^2 was 64 times greater than σ_n^2.

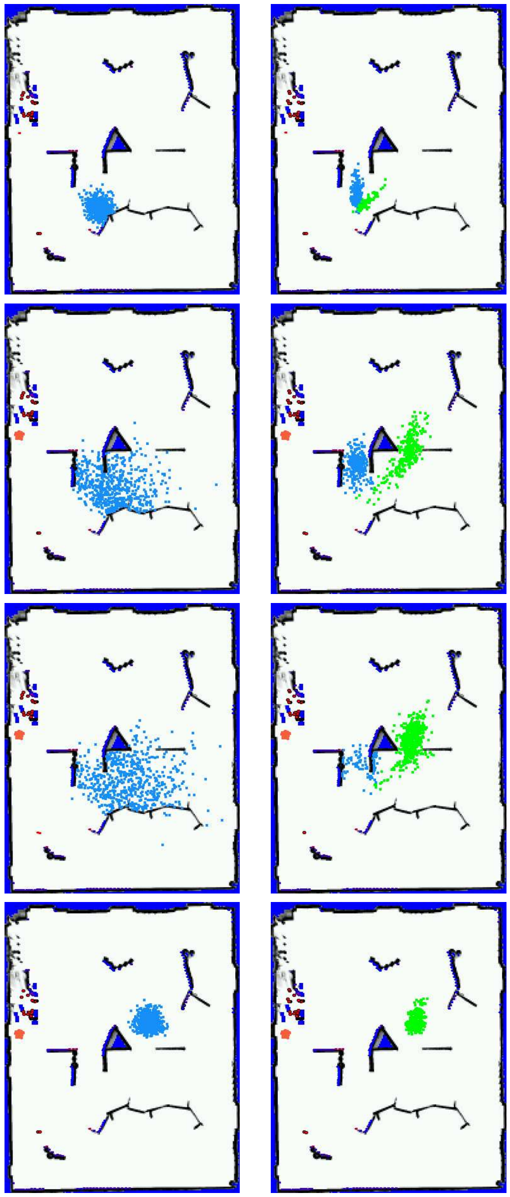

Fig. 1. Tracking a person during the occluded part of their path using the Brownian motion model (left) and the plan-based motion model (right). From top to bottom: just after losing sight of the person; the Brownian filter's variance starts to grow; the Brownian filter fails to predict that the person is about to emerge from occlusion; the Brownian filter's variance remains higher for several steps after reacquiring the person.

$$
\begin{aligned}
\theta &= \text{atan2}(l_{\text{nearest}_x}, l_{\text{nearest}_y}) \\
d &\sim \text{abs}(N(0, \sigma_d^2)) \\
n_x, n_y &\sim N(0, \sigma_n^2) \\
x_t &= x_{t-1} + d * \cos(\theta) + n_x \\
y_t &= y_{t-1} + d * \sin(\theta) + n_y
\end{aligned}
$$

There are a couple of issues to note when implementing the plan-based motion model. The model will only perform well when the set of goals provide a reasonable "coverage"

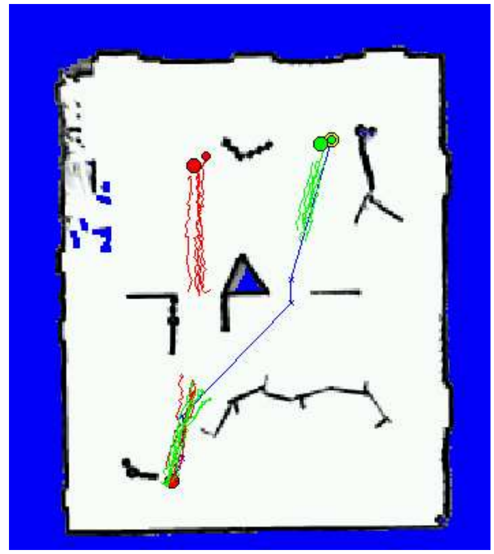

Fig. 2. The optimization process. The large circles are the current best goals and the small circles represent proposed goals. The clusters of training trajectories are shown with lines drawn between points at which the person was observable. A plan is shown as a series of line segments with x's at their endpoints. Obstacles in the map are shown in black.

of the paths people follow in the space. This motion model makes stronger assumptions about a person's dynamics than a Brownian motion model. If these assumptions are incorrect, tracking performance could be worse rather than better, though the effects might not be apparent unless there are periods of occlusion of the person being tracked. It is therefore reasonable to expect the motion model to perform well only in areas for which training examples are available. Also, because a path planner is used, we implicitly assume that people follow reasonably efficient paths to their goals. If this is not the case (for example, if people change direction to avoid an obstacle that cannot be detected by the laser scanner), the motion model will yield poor predictions.

IV. Learning Goals

In order to learn the goals, it is necessary to collect training data. The training data consists of trajectories of people moving through the area that the robot will be tracking in. These trajectories are a series of point estimates of a person's location at the time of each laser scan. We obtained the trajectories by using the Brownian motion model tracker on pre-recorded laser data. These trajectories are then clustered into groups, with each group following roughly the same path. For the purposes of our experiment the trajectories were clustered by hand, but clusters could also be obtained using the method described in [6]. We also require an occupancy grid map of the area, which is used by the tracker to recognize expected laser readings and by the path planner to determine what locations are impassable.

The goal locations are initialized randomly. The planner returns a plan from the start of a trajectory to the proposed goal location for its cluster. We used an MDP planner (described in

[7]), but the choice of path planner is not of great importance. Each trajectory in a cluster is scored according to the distance between each of its points and the nearest line segment of the path returned by the planner. The constraint is added that it is not possible to “go backwards” along the path returned by the plan: a point cannot be compared to any line segments earlier in the plan than the segment that the point before it in the trajectory was compared to. Figure 2 shows a snapshot of the learning process using the training data from our experiment.

The score for the trajectory is the sum of the squared distances between the points and the plan, normalized by the number of points. The distance is normalized so that trajectories which contain more sample points (because the person was moving more slowly) will not be weighed more heavily than other trajectories. The score of a cluster is the sum of the scores of its trajectories. Because it is difficult to find the gradient of the plan with respect to changes in the goal, the optimization proceeds by hillclimbing. New goal positions are proposed by adding a 2D gaussian offset to the best cluster goal found so far. Each time that the score is improved, the new goal location is accepted and the variance of the next proposed offset is reduced.

Computing the likelihood of a particular goal location exactly would involve integrating the position of the points along the path out of the posterior. However, this computation is difficult to do, and we don’t believe that it would change the solution enough to significantly improve the performance of our tracking algorithm. So, we perform three approximations which greatly simplify our computation. First, we make a most-likely-point approximation to the integral: instead of optimizing

$$P(\Theta \mid X^n) = \int P(\Theta, D^n \mid X^n) dD^n$$

we use instead

$$\max_{D^n} P(\Theta, D^n \mid X^n) = \max_{D^n} P(\Theta \mid D^n, X^n) P(D^n \mid X^n)$$

where X^n are all the data points of the n trajectories in the training set, D^n are the points along the planned path that each data point corresponds to, and Θ is the set of goal locations.

Second, because the maximization over hidden variables would be time-consuming to solve exactly, we make a simple myopic approximation to the most likely path: we project each observation onto the path, then move it forward if it falls behind the previous one. Finally, after setting the hidden variables we neglect the term $P(D^n \mid X^n)$ since it is usually nearly constant. These approximations lead to a optimization procedure that quickly and cheaply finds goal locations that are adequate for our purposes.

V. Experiment

To verify the quality of our approach, we constructed a simple yet realistic test case. We set up an environment in which people start from one location and then choose to pass through one of two doors. A robot observes their movements using a laser scanner, but the part of the room immediately before the doors is blocked from its view (Figure 2). Seventeen trajectories of were collected, with eight trajectories passing through the left door and nine through the right. They were divided into eleven training examples that were used to learn the goal locations and six examples (three left, three right) that were used as the test set. We compared the performance of a particle filter with a Brownian motion model to our plan-based motion model. Each particle filter used 600 particles to track a person. Our motion model performed well with fewer particles, but that number was necessary to ensure good coverage of the unobservable areas by the Brownian motion model when the view of the person was occluded.

A. Computational Considerations

The learning process for the goals converges quickly for our experiment, usually within around 200 iterations, which takes under a minute to run with a visualization of the optimization process on a 700 megahertz Pentium III. We determined the convergence of the optimization manually by inspection. But it would be simple to stop the process automatically by setting a threshold on the number of iterations without improvement or on a value of the score for goal locations that is an acceptable amount of error. The speed of the optimization we achieved by making simplifying assumptions is likely to be more beneficial for more complex environments with a larger number of goals and trajectories. In these situations, an EM-based optimization algorithm would become very computationally expensive. We also expect that random restarts might be needed to achieve good solution quality with more complex training data, though they were unnecessary for our experiment.

Both versions of the particle filter implemented for the experiment in this paper run in real time with laser updates occurring at the rate of 12 Hz. The plan-based motion update runs approximately twice as slow as the Brownian motion update on a filter with the same number of particles. But it is important to note that its error (as described in Section VI) is much smaller. In order to obtain the same error rate as with the Brownian motion model, the tracker using the plan-based model could use fewer particles. In fact, using half as many particles with the plan-based tracker yields lower error than the Brownian motion based tracker with 600 particles, so the plan-based motion model is faster for the same performance because it can track with fewer particles.

Considering only the time needed for the motion update ignores an aspect of the plan-based motion model that also adds to its execution time, the time it takes to plan. In our implementation of the algorithm, we were able to replan to all the goals each time a person was observed. However, this may not be possible as the number of goals or the complexity of paths increases. In this case, plans could be updated less frequently, or a plan or representative set of plans could be computed and cached prior to tracking.

VI. Results

In Figure III-B, the filters are shown at various stages of tracking a person going through the right doorway. The figures

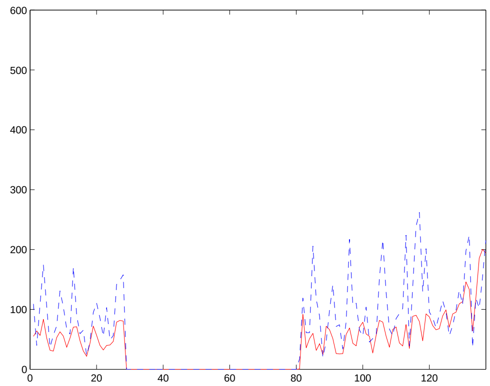

Fig. 3. The particle filter normalization constants at each timestep of a trajectory. The part of the graph in the middle with zero weight is the period of time when the person was occluded by the wall.

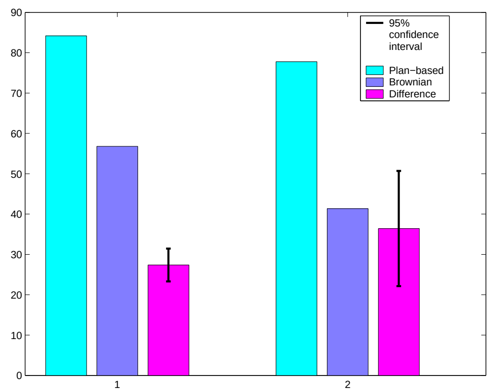

Fig. 4. The particle filter normalization constants. The first group of bars is the average over all timesteps when the person was observed, except the first 2 time steps after the person was reacquired. The second group is the average over the 2 time steps after reacquisition. Higher normalization constants correspond to smaller tracking error.

show how the plan-based motion model focuses the particles in the more likely parts of the space. In the pictures of the plan-based filter, the shade of a particle indicates which goal its motion is determined by.

On average, our motion model performed better overall than the Brownian motion model. The advantage of our approach is most pronounced for the first two timesteps when a person is observed again after the period of occlusion. This is because the motion models have had many timesteps to project the particles forward in time without being "corrected" by the perception update. It can be seen in Figure 1 that our motion model focuses the particles in the locations where the person is most likely to actually be while they are occluded, unlike the Brownian motion model, which just disperses its particles evenly over the space.

We measured the performance by analyzing the differences between the normalization constants, described in Section III-A, for each filter at each timestep. The normalization constant was Figure VI shows the normalization constants of each filter plotted over the duration of a single example trajectory. These weights are a measure of the performance of the filter: they are estimates of the probability of the most recent observation given the previous state, so filters that do better at predicting actual observations will produce larger weights.

The results of the experiment are summarized in Figure 4. The first group of bars shows the average normalization constant for the plan-based and Brownian motion model and the average difference between the normalization constants of the two filters. The averages were computed over each timestep that a person was observed, except for the first two timesteps when he or she was reacquired after the period of occlusion. The second group of bars shows the same averages for only the first two timesteps when a person had been reaccquired. A matched t-test was performed on the averages for each group. The differences in the means of both groups were determined to be significant, at $p < 3 \times 10^{-16}$ for the first group and $p < .0002$ for the second group. A 95% confidence interval on the difference of the means is shown on the graph. These results show that the plan-based motion model performs significantly better than the brownian motion model, both overall and immediately after reacquisition. The results suggest that the benefit in performance is larger right after reacquisition, but this comparison is not statistically significant.

VII. Conclusion

In this paper, we proposed a motion model for people tracking that is inspired by the goal-oriented nature of people's movement. This motion model involves a learning component that allows it to use information about people's common trajectories in a specific environment to learn goal locations. The goal locations are optimized so that paths produced by a planner agree well with the training trajectories. Paths planned from the location of a person being tracked to these goals are used by the motion update to project the hypotheses forward in time. We compared the performance of our motion model to a simple Brownian motion model within the framework of a particle filter based people tracker. Experimental results verified that our motion model performed better, creating a more realistic distribution over positions.

Acknowledgement

Thanks to Matt Rosencrantz for his help with deciphering the internal workings of CARMEN.

References

[1] S. T. M. Montemerlo and W. Whittaker, "Conditional particle filters for simultaneous mobile robot localization and people-tracking," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2002. [Online]. Available: citeseer.nj.nec.com/thrun02conditional.html

[2] E. Meier and F. Ade, "Using the condensation algorithm to implement tracking for mobile robots," in *Third European Workshop on Advanced Mobile Robots, Eurobot99*, 1999, pp. 73–80. [Online]. Available: citeseer.nj.nec.com/meier99using.html

[3] B. G. F. M.Berna, B.Lisien and S.Thrun, "A learning algorithm for localizing people based on wireless signal strength that uses labeled and unlabeled data," in *International Joint Conference on Artificial Intelligence*, 2003.

[4] J. H. H. K. L. Liao, D. Fox and D. Schulz, "Voronoi tracking: Location estimation using sparse and noisy sensor data," in *Proceedings of the Conference on Intelligent Robots and Systems (IROS)*, 2003.

[5] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for on-line non-linear/non-gaussian bayesian tracking," vol. 50, no. 2, pp. 174–188, 2002. [Online]. Available: citeseer.nj.nec.com/maskell01tutorial.html
[6] M. Bennewitz, W. Burgard, and S. Thrun, "Learning motion patterns of persons for mobile service robots," in *IEEE Conference on Robotics and Automation*, 2002. [Online]. Available: citeseer.nj.nec.com/bennewitz02learning.html
[7] N. Roy and S. Thrun, "Motion planning through policy search," in *Proceedings of the Conference on Intelligent Robots and Systems (IROS)*, Lausanne, Switzerland, 2002.