Temporal Persistence Modeling for Object Search

Russell Toris\(^1\) and Sonia Chernova\(^2\)

Abstract—We present a novel solution to the object search problem for domains in which object permanence cannot be assumed and other agents may move objects between locations without the robot’s knowledge. We formalize object search as a failure analysis problem and contribute temporal persistence modeling (TPM), an algorithm for probabilistic prediction of the time that an object is expected to remain at a given location given sparse prior observations. We show that probabilistic exponential distributions augmented with a Gaussian component can accurately represent probable object locations and search suggestions based entirely on sparsely made visual observations. We evaluate our work in two domains, a large scale GPS location data set for person tracking, and multi-object tracking on a mobile robot operating in a small-scale household environment over a 2-week period. TPM performance exceeds four baseline methods across all study conditions.

I. INTRODUCTION

The ability to effectively locate objects is critical for robot tasks across a wide range of assistive, household and service applications. Commonplace tasks, such as pick and place or manipulation, require the robot to navigate to different locations to retrieve objects, often given only uncertain information about the object’s current location. Within this context, the problem of active visual search (AVS) is that of effectively predicting object locations in a large-scale environment using primarily visual sensing [1], [2].

Recent work in semantic mapping has explored a wide range of AVS solutions, considering search for both known [3], [4] and previously unknown [5], [6] objects. Many of the presented techniques leverage semantic information, such as object co-occurrence probabilities [5] or typical object locations [1]. One limitation of existing methods is that they typically operate under the assumption of object permanence; in other words, once found, objects are assumed to remain in their relative position.

In this work, we address the problem of AVS within a domain where other agents (e.g., people) may move objects around without the robot’s knowledge. For example, in a household setting, a person getting a snack is likely to carry a plate from the kitchen to the coffee table, and then back half an hour later. Similarly, in an office setting, shared objects such as staplers or computer accessories are often moved between work areas. We are interested in the problem of effectively predicting the permanence of an object at a given location in order to enable improved visual search.

II. BACKGROUND AND RELATED WORK

At its core, semantic mapping deals with mapping human spatial concepts to objects within the world [8]. Many approaches have been developed for tackling this broad research problem within the robotics and perception communities [9], [3], [8], [10], [11]. Seminal work by Pronobis breaks down objects to be what he refers to as “spatial concepts” [8], which can be observational properties (e.g., the object is green) or relationship properties (e.g., the object is in the kitchen). Thus, at its core, the problem of semantic mapping is that of assigning some type of semantic (e.g., human-readable and understandable) label to a spacial entity within the world.

Semantic map information can also be used to determine object placement. For example, Mason and Marthi [3] build a semantic world model that enables the detection and tracking of objects over time, resulting in the ability to model where objects are in the world (e.g., where did you see object $x$), or how objects move around (e.g., where do instances of $x$ go throughout the day). The major limitation of the proposed system is that it requires extensive time for post-processing.

In an alternate approach to object search, Kollar and Roy [5] predict the location of previously unseen objects given the probability of their co-occurrence with other known objects in the environment. This approach assumes a map and the locations of known objects in the environment are given to the robot. Then, given a novel object that has not been previously observed in the environment, the system leverages an online object co-occurrence database derived from Flickr to model the co-occurrence probabilities used to determine the likely new object location based on what it already knows about the environment. The work presented in our paper is complimentary to Kollar and Roy’s approach in that it...
seeks to generate and maintain the model of known object locations, which may then be used to predict the location of novel objects through their co-occurrence method.

In [6], [12], the authors present an approach that enables the robot to search the web in order to identify an object’s most likely location. The approach, called ObjectEval, searches the Web to infer the probability that an object, such as coffee, can be found in a location, such as a kitchen. The algorithm is able to dynamically instantiate a utility function for different locations using this probability, with the benefit that the robot is able to effectively search for arbitrary objects is may never have previously encountered. Again, we view this work as complimentary to the contributions of our paper; the technique presented in our work utilizes memory and online learning to track the locations of known objects, as opposed to focusing on the ability to find novel objects. Once objects are known, recognized and seen multiple times, it is more effective for a robot to track their likely location in a particular environment than to rely on a general notion of likely locations mined from the web, since the web-based suggestions will perform well only in domains that fit common stereotypes.

Another popular approach to solving the object search problem involves augmenting the environment with additional sensors. Most popular is the use of radio-frequency identification (RFID) tags attached to key objects and reference points. For example, in [13], Deyle et al. enable a PR2 robot to locate objects in a highly-cluttered household environment by placing ultra-high frequency radio-frequency identification (UHF RFID) tags on objects. In their work, the problem of inferring object poses and the variability in signal strength is addressed in order to robustly search for a wide site of objects. However, even with the use of RFID tags, an object search model is still required to enable the robot to determine which area to scan for objects.

Finally, Lorbach et al. [4] present an approach in which the robot navigates around the environment and constructs a scene graph from its observations both at a semantic and spatial level. The robot then uses this information to develop a series of knowledge bases: the Scene Structure which contains the observed scene graph, the Domain Knowledge containing co-occurrence information (similar to the work in [5]), Physical Constraints which encode information such as large objects not being inside of smaller objects, Logical Consistency which ensures an object can only be in one place at a time, and Search History which serves as short term memory for spatial relationships. While this approach begins to explore the idea of history in guiding its search, we argue that simply remembering the past observations ignores the inherent temporal information about the object (i.e., how long it will remain there). This idea will be explored explicitly in our analysis. Furthermore, the need for hand-coded domain knowledge (which the authors say is to remove noise) makes such an approach less general.

### III. Temporal Persistence Modeling

In this work, we contribute temporal persistence modeling (TPM), an algorithm for probabilistically modeling object locations based on sparse temporal observations. We define temporal persistence as the time an object is expected to be in a particular location before moving to another location (i.e., “disappearing”). More formally, let us consider the case of searching for an object \( o \in O \). Given the current time \( t_c \), and the last time object \( o \) was seen at each location \( l \in L \), denoted \( t_{o,l} \), our goal is to calculate the probability that \( o \) is currently located at \( l \).

To address this problem, we leverage concepts from the fields of reliability theory [7] and failure analysis [14]. Widely applied in biomedical studies, industrial life testing, psychology, economics and engineering disciplines, reliability theory is a branch of statistics used for modeling the expected duration of time until a particular event occurs, such as death in biological organisms and failure in mechanical systems.

Within the context of our work, the event time distribution we are interested in modeling is the time at which an object is removed from one location and placed at another. We formalize the object search problem using reliability theory as a maximization of the probabilistic temporal persistence \( P_{tp} \) equation:

\[
\arg\max_{l \in L} P_{tp}(o \text{ located-at } l|t_c, t_{o,l})
\]  

We now present how \( P_{tp} \) is derived. The first step in defining our temporal persistence model utilizes the exponential distribution [15]. In general, exponential distributions model the time between events and contain a single rate parameter, \( \lambda \). In order to accurately define \( \lambda \), we utilize the fact that the mean \( \mu \), or expected value of the distribution, is defined as \( \mu = \frac{1}{\lambda} \). In TPM, \( \mu_{o,l} \) represents the average or expected time that elapses between the last known observation of object \( o \) at \( l \), and the time at which \( o \) is removed from \( l \). We note that a problem which must be addressed is that observations at the actual time of removal are unlikely. That is, as the robot goes around the environment making observations, it will most likely not observe the actual time an object moves locations (unless, of course, the robot moves the object itself). Such an assumption would require a global visual sensing system, a case which we assume is not available. We will later address how to probabilistically sample and calculate \( \mu_{o,l} \), but for now we assume the value is known.

Given our exponential model for a given object-location pairing \( \exp(\lambda)_{o,l} \) with associated expected value \( \mu_{o,l} \), we can calculate the probability that \( o \) has been removed from \( l \) on or before the current time \( t_c \) from the cumulative distribution function (CDF) given in:

\[
cdf_{\exp}(t, \lambda) = \begin{cases} 
1 - e^{-\lambda t} & t \geq 0 \\
0 & t < 0 
\end{cases}
\]

Therefore, to answer our initial question, the probability that \( o \) is still located on \( l \) at time \( t_c \) is analogous to 1 minus
the probability $o$ has been removed from $l$ at or before $t_c$. Our equation for this persistence model is given in Equation 3; a visualization is also provided in Figure [1]

$$P_{tp}(o \text{ located-at } l | t_c, t_o, l) = 1 - \text{cdf}_{exp}((t_c - t_o, l), \frac{1}{\mu_{o,l}}) = e^{-\frac{1}{\mu_{o,l}}(t_c - t_o, l)}$$

(3)

The final missing component of the equation is determining $\mu_{o,l}$ for each model. As stated earlier, if we could assume a large-scale continuous perception system in the environment, we could assume we know exactly when $o$ was removed from $l$; however, if this assumption were true, our system would have perfect information about object locations, eliminating the need for this entire approach. Since such a system is unlikely to be widely available, is currently infeasible, and is costly to deploy, we assume that no such data is available.

Instead, we assume that the robot will periodically record observations at random time steps throughout the day as it goes about its daily tasks and duties. It is also possible to program the robot to perform regularly scheduled sweeps of the environment to observe object locations. However, such behavior could be distracting and intrusive to other occupants of the space, and so we do not rely on such functionality in our system. Instead, we perform TPM updates each time the environment to observe object locations. However, such behavior could be distracting and intrusive to other occupants of the space, and so we do not rely on such functionality in our system. Instead, we perform TPM updates each time the robot makes a new observation of location $l$ which is denoted as $\Omega$. This set $\Omega$ contains a list of all objects from $O$ which are observed on location $l$. At the end of the algorithm, the new value of $\mu_{o,l}$ for each object-location pairing allows us to create our temporal persistence models and therefore we now have enough information to maximize Equation [1] during object search.

In order to generate our estimated time of removal, we fit a Gaussian distribution over the time frame of $\Delta_t$, and randomly choose a value from inside that distribution as a sample for updating $\mu_{o,l}$. Note an alternate distribution, such as the uniform distribution, may be used instead of the Gaussian.

The complete algorithm for updating $\mu_{o,l}$ in TPM is shown in Algorithm [1]. The algorithm is called each time a robot makes a new observation of a location $l$ which is denoted as $\Omega$. This set $\Omega$ contains a list of all objects from $O$ which are observed on location $l$. At the end of the algorithm, the new value of $\mu_{o,l}$ for each object-location pairing allows us to create our temporal persistence models and therefore we now have enough information to maximize Equation [1] during object search.

In this section, we present analysis of our temporal persistence modeling approach in two applications, 1) large-scale tracking of a single target based on real-world GPS cell phone data, and 2) multi-object tracking using a mobile robot over a 2-week period in a small-scale household setting. In both applications we compare the performance of TPM
Algorithm 1 Temporal Persistence Modeling μ Update

\[
\begin{align*}
l & \leftarrow \text{location being observed} \\
t_c & \leftarrow \text{current time} \\
\Omega & \leftarrow \text{makeObservation}(l) \\
\text{for each} & \text{ object } o \text{ observed in } \Omega \text{ (i.e., } \forall \ o \in \Omega) \text{ do} \\
\text{ if } o & \text{ was not previously observed at } l \text{ then} \\
\text{ mark } o & \text{ as observed at } l \text{ at time } t_c \\
\text{ else} & \\
\text{ mark } o & \text{ as still observed at } l \text{ at time } t_c \\
\text{ end if} \\
\text{ end for} \\
\text{for each} & \text{ object } o \text{ not observed in } \Omega \text{ (i.e., } \forall \ o \in \{O \setminus \Omega\}) \text{ do} \\
\text{ ts } & \leftarrow \text{first observation of } o \text{ at } l \\
\text{ tf } & \leftarrow \text{latest observation of } o \text{ at } l \\
\mu & \leftarrow \{\mu_1, \ldots, \mu_n\} \text{ previous estimates of } \mu_{o,l} \\
\mu_{o,l} & \leftarrow (\sum_{j=1}^{n}(1/n)\mu_j)+tf \\
\text{ mark } o & \text{ as not observed at } l \\
\text{ end for}
\end{align*}
\]

in predicting the location for a given object \( o \) against four baseline methods:

1) TP Random: We select a random value for \( \mu_{o,l} \) and then apply the remainder of the TPM algorithm as described above. The \( \mu_{o,l} \) value is chosen in the range from the minimum to the maximum time difference between two observations in the training set. This experimental condition aims to show that the \( \mu \) value within TMP is intelligently chosen, as well as the impact of a poor value.

2) Last Seen: We predict the location for \( o \) based on the last location where \( o \) was observed.

3) Most Frequent: We predict the location for \( o \) based on that object’s most frequently observed location.

4) Random: We randomly select a location for \( o \) from among the known locations.

A. Single Object Tracking from GPS Data

In our first experiment, we utilize GPS tracking data from the Microsoft Research GeoLife dataset to verify the ability of our temporal persistence models for learning probable object locations based on sparse temporal observational data. A GPS trajectory of this dataset is represented by a sequence of time-stamped points, each of which contains the latitude, longitude and altitude of a single cell phone user. The recordings span the full range of activities of everyday life, including regular work commutes, as well as entertainment and sports activities, such as shopping, sightseeing, dining, hiking, and cycling, making it an compelling dataset for temporal tracking.

For the purposes of our work, we are interested in modeling the temporal persistence of each individual cell phone user. Specifically, given a sparse set of observations of prior user locations, we seek to predict the user’s current location. It is important to note that, when dealing with people, context aware methods are likely more suitable for location tracking than temporal persistence modeling. Information such as a person’s place of work, members of their social circle, favorite places to eat, or even the current direction of motion are helpful in aiding in the analysis and prediction of a person’s activities. In our work, however, we are primarily interested in tracking inanimate objects, and thus we utilize the location information in this dataset as a useful benchmark due to its size and diversity of users. As we show in the results section, temporal persistence modeling performs extremely well on this dataset, even without the use of context information.

1) Experimental Setup: For our analysis we discretize the user trajectory data by superimposing a uniform \( 10 \times 10 \) grid over the range of latitude and longitude values for a given user. Therefore, each cell in the grid becomes a “location” and the user becomes the “object” for which we want to predict the location. Trajectories within the GeoLife dataset are logged in a dense representation, every 1-5 seconds. We subsample this data in order to create a relatively sparse training and test set that more closely resemble the frequency of object location observations a robot is likely to obtain during typical operation. The test set is generated by sampling 10% of a given user’s data uniformly and without replacement. We then sort the resulting data points according to their timestamps and train either TPM or one of the baseline methods (in the case of TPM, we iteratively train the model using Algorithm 1). We generate the test set by sampling 10% of the remaining data from that user, again uniformly and without replacement. Each data point in the test set is used to query the trained model (in the case of TPM, by applying Equation 1). We then compare the returned value to the ground-truth of the actual discretized location to obtain an accuracy measure for the approach.

Note that throughout the verification process, we update the last seen time for the temporal persistence models \( t_{o,l} \) and last seen location for the last-seen method as verification observations are made. Further note that this is not the same as updating the TPM parameter \( \mu_{o,l} \).

2) Results: We report evaluation results for 10 users randomly selected from the GeoLife dataset. Figure 4 presents the average prediction success rate for all 10 users across all five algorithms. Due to the nature of the GeoLife dataset, the number of datapoints available for each user varies greatly; the \( N \) value next to each user label in the figure indicates the number of training/testing samples available per user (i.e., 10% of the entire dataset for that user). All performance values are averaged over 50 trials per user, with training and testing samples randomly drawn for each trial. Points along the outside of the graph indicate performance close to 100%, whereas points near the center of the graph indicate predictive performance close to 0%.

As can be seen in the graph, the performance of all baseline methods varies greatly across users, while the
Fig. 4. Performance of TPM and four baseline methods in predicting the location of 10 independent users in the GeoLife dataset.

The performance of TPM (dashed blue line) consistently outperforms all other methods for all 10 users. Based on the data findings, we can observe certain patterns among the user data. Users 1, 3 and 5 appear to be sedentary, remaining in the same place for long periods of time. This can be observed from the fact that both the Last Seen and Most Frequent baselines perform very well for these users. The remaining users change location to varying degrees, with User 9 exhibiting the least repetitive behavior. The Last Seen metric performs well across the full range of users. This is to be expected since multiple consecutive readings typically fall within the same discrete grid cell of our discretized state space. However, TPM dominates the performance of Last Seen (and all other baselines) with statistical significance of $p \leq 0.001$ as calculated by a $t$-test performed against each method independently. Additionally, we note that TP Random performs significantly more poorly than the complete TPM algorithm, indicating that customizing $\mu_{o,l}$ with the respect to an individual user’s behavior significantly improves the performance of the algorithm.

B. Multi-Object Tracking in a Robotic Apartment Environment

The GPS data evaluation above showed that TPM can effectively be used to track a single object (in that case, a person) over time. In this evaluation we evaluate our approach in tracking multiple physical objects in the context of a real-time, real-world robotics domain. We utilize a mobile manipulation robotic platform operating in a small-scale household environment consisting of a kitchen area, dining area and a seating area. Figure 5 presents an abstract map of the physical space, and Figure 6 shows the robot located in the kitchen area. Within the context of this experiment the robot does not manipulate any objects, but it is able to navigate the space and perform object recognition [17] to identify the objects of interest. The robot is provided with a map of the space that contains semantic labels associated with different locations in that space (e.g., kitchen table, coffee table, couch). Beyond that, we assume the robot has no additional information about any prior object locations when it begins.

1) Experimental Setup: The dataset for this experiment was collected over a 2-week period, during which the robot was operated 6 days per week, 5 hours a day. The time restriction is due in part to the robot’s battery limitations, but it also represents a reasonable estimate of the time a mobile robot may actively operate inside a typical household setting. During its operation time, the robot adhered to the following procedure:

1) Select a target location $l \in L$ uniformly at random.
2) Navigate to the selected location and make an observation of all objects located on that location.
3) Update TPM as defined in Algorithm 1.
4) Select a random wait time from a uniform distribution in the range of $[1, 20]$ minutes; return to Step 1.

This behavior pattern was selected because it mimics an observation pattern the robot might obtain while going about daily tasks in a household environment. For example, the robot may go to tidy up the coffee table and in the process observe the objects that are currently placed there. Then it may go to the kitchen to clean up the dishes and in the process observe the objects placed there. Thus our work specifically seeks to target the scenario in which the robot makes incidental observations over the course of its normal operation, as opposed to a targeted sweeping search that records the location of every object in the environment. In this study the robot does not move any objects between locations itself, modeling object movement in that scenario would simply require a straightforward update to a world model or semantic map. However, TPM does support this scenario; if the robot were to move an object, the sample value to add to $\mu_{o,l}$ would be explicitly known (since

---

Given that the user has significantly less data than all other users, it is likely that we simply do not get the chance to observe the person’s behavior long enough to see repeated patterns. One of the strengths of TPM is that its model can be constructed from relatively little data, which will be discussed further during the second experiment.
the robot knows the exact time it moved the object), and therefore it would not need to be sampled from the Gaussian distribution.

In this paper, we are instead interested in the more challenging case in which objects are moved by other agents (in our case, people) in the environment. We used the following set location locations in the experiment \( L = \{TV \text{ stand, arm chair, shelf, couch, coffee table, kitchen counter, kitchen table} \} \). The objects tracked in the study included \( O = \{\text{book, cup, plate, fork, candle, potted plant}\} \). Each of the objects was used to evaluate a different temporal persistence behavior:

1) **Regular Alternating 1 (RA1, plate)** The location of the plate alternated between the kitchen table and counter. The plate was on the kitchen counter for the first 3 hours of the day, then one hour at the kitchen table before being returned to the counter for the remainder of the day.

2) **Regular Alternating 2 (RA2, fork)** The location of the fork alternated between two hours on the kitchen counter and 30 minutes on the coffee table.

3) **Regular Alternating 3 (RA3, candle)** The location of the candle alternated between one hour on the coffee table and one hour on the kitchen table.

4) **Pattern Change (PC, book)** During the first week, the book was located on the kitchen table for the first hour of the day and then moved to the bookshelf for the remainder of the day. During the second week, the book remained on the TV unit stand for the first 3 hours of the day before moving to the bookshelf for the remainder of the day.

5) **Stationary (S, plant)** The plant remained on the bookshelf throughout the two-week study.

6) **Random (R, cup)** The location of the cup was changed at random over the course of the study by selecting \( l \in L \) uniformly at random, and selecting a time period uniformly from the range \([1, 20]\) minutes, then repeatedly moving the cup to the selected location after the designated period of time elapsed.

Each of the above objects can be viewed as a different study condition, ranging from random movement to stationary objects. RA 1-3 represent the most interesting temporal persistence variant for our target use case, one in which objects are moved based on some implicit customs or patterns that are unknown to the robot but that it is able to indirectly observe. The PC condition was selected to evaluate the scenario in which the use pattern of an object changed significantly during the period of observation.

In total, our 72 hours of robot operation included 412 observation snapshots taken across the seven locations in the domain, resulting in 506 object observations. Below, we use the resulting dataset of timestamped observations to evaluate the robot’s ability to effectively predict the location of an object.

2) **Results:** In order to evaluate the performance of TPM, we train a temporal permanence model for each object based on the observations obtained during the two-week data collection period. We then generate a test set by sampling an equal number of data points over a future two week period, following the same object placement schedule as previously described. The ground truth of each data point in the test set is determined based on the originally defined schedule.

Table [I] presents the evaluation results for each of the study conditions; \( N \), the number of observations of each object, is also provided. As can be seen in the table, the performance of TMP dominates or matches the performance of the baseline methods for all six study conditions. RA1 and RA2 behave similarly in that both have a single location where the object of interest remains for the majority of the time; thus, both Last Seen and Most Frequent is its ability to predict that the object has most likely been moved from its last seen location after some period of time elapses.

RA3 helps to demonstrate the benefit a temporal model has over the heuristic baselines. In this study condition, the object alternates between two positions, spending equal time at both. As a result, both Last Seen and Most Frequent are able to guess the correct location of the object with only 50% accuracy, while TPM is again able to predict the object’s movement to a greater degree. By adding a temporal decay to the prediction strategy, the robot is able to gain an improvement in its success rate. However, this condition exemplifies a difficult case even for TPM. Consider the following:

Note that each TPM is independent of the others. Assume that both the locations \( l_1 \) and \( l_2 \) in the alternating case example have perfect \( \mu_{o,t} \) values each of 1 hour. Now assume we observed \( o \) on \( l_1 \) 2 hours ago and \( o \) on \( l_2 \) 1:59 ago. Without any higher-level reasoning or knowledge of the other locations, the robot will pick \( l_2 \) as the location regardless of the fact that it has most likely switched. This is a direct result of \( \mu_{o,t} \) being the same for each location. Note that, however, by increasing the observation frequency at each location we could improve the success rate of this case.

The PC condition is similar to RA1 in that the object
The observed trend indicates that the performance of TPM is relatively low early on, but that $\mu_{o,l}$ will converge quickly once it has enough data to come close to its optimal value. Given that $\mu_{o,l}$ is calculated based on a weighted average, more data will simply serve to strengthen (i.e., stabilize) $\mu_{o,l}$, not necessarily provide a better estimate. For example, with the plate (RA1), early observations gave reasonable estimates of the kitchen counter leading to the initial boost in performance, but once it was able to reason

<table>
<thead>
<tr>
<th>Table I</th>
</tr>
</thead>
<tbody>
<tr>
<td>Results from the Real-World Robotic Experiment.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>RA1 (N = 89)</th>
<th>RA2 (N = 89)</th>
<th>RA3 (N = 79)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TPM</td>
<td>96.360%</td>
<td>83.021%</td>
<td>57.362%</td>
</tr>
<tr>
<td>TP Random</td>
<td>93.187%*</td>
<td>75.932%*</td>
<td>52.861%*</td>
</tr>
<tr>
<td>Last Seen</td>
<td>96.044%</td>
<td>80.114%</td>
<td>50.460%</td>
</tr>
<tr>
<td>Most Frequent</td>
<td>96.097%</td>
<td>80.166%</td>
<td>49.539%</td>
</tr>
<tr>
<td>Random</td>
<td>18.459%</td>
<td>19.678%</td>
<td>18.762%</td>
</tr>
<tr>
<td></td>
<td>PC (N = 86)</td>
<td>S (N = 78)</td>
<td>R (N = 87)</td>
</tr>
<tr>
<td>TPM</td>
<td>92.741%</td>
<td>100.000%</td>
<td>24.071%</td>
</tr>
<tr>
<td>TP Random</td>
<td>28.021%*</td>
<td>17.970%*</td>
<td>21.652%</td>
</tr>
<tr>
<td>Last Seen</td>
<td>91.250%</td>
<td>100.000%</td>
<td>17.804%</td>
</tr>
<tr>
<td>Most Frequent</td>
<td>91.250%</td>
<td>100.000%</td>
<td>14.945%</td>
</tr>
<tr>
<td>Random</td>
<td>20.895%</td>
<td>21.014%</td>
<td>19.445%</td>
</tr>
</tbody>
</table>

*Indicates an average of 10 runs each with a different random value for $\mu_{o,l}$.

In summary, this paper contributes a novel temporal persistence modeling algorithm that enables a robot to model effectively and predict likely object locations in its environment. Unlike past approaches in the areas of both active visual search and semantic mapping, we focus solely on temporally based relationships between objects and search locations. Our approach builds probabilistic exponential models in order to model when an object will no longer be located on a given location given the last known time the object was observed on that location. In doing so, we compute a weighted average guess on what the rate parameter $\mu_{o,l}$ of the models are by sampling from a Gaussian distribution over the last known observation time and the current time at which the object was no longer seen at the location.

We evaluated our approach using two data sets, evaluating both single-object tracking in a large-scale GPS location data set, and multi-object tracking in a real-world robotics deployment lasting two weeks. The performance of our algorithm dominates that of four baseline methods in both domains, and shows the importance of correctly estimating the $\mu_{o,l}$ parameter.

V. Conclusion

In summary, this paper contributes a novel temporal persistence modeling algorithm that enables a robot to model effectively and predict likely object locations in its environment. Unlike past approaches in the areas of both active visual search and semantic mapping, we focus solely on temporally based relationships between objects and search locations. Our approach builds probabilistic exponential models in order to model when an object will no longer be located on a given location given the last known time the object was observed on that location. In doing so, we compute a weighted average guess on what the rate parameter $\mu_{o,l}$ of the models are by sampling from a Gaussian distribution over the last known observation time and the current time at which the object was no longer seen at the location.

We evaluated our approach using two data sets, evaluating both single-object tracking in a large-scale GPS location data set, and multi-object tracking in a real-world robotics deployment lasting two weeks. The performance of our algorithm dominates that of four baseline methods in both domains, and shows the importance of correctly estimating the $\mu_{o,l}$ parameter.
REFERENCES


