GATMO: a Generalized Approach to Tracking Movable Objects

Garratt Gallagher and Siddhartha S. Srinivasa and J. Andrew Bagnell and Dave Ferguson

Abstract— We present GATMO (Generalized Approach to Tracking Movable Objects), a system for localization and mapping that incorporates the dynamic nature of the environment while maintaining semantic labels. Objects in the environment are broken down into multiple mobility levels, from static (walls) to highly mobile (people), by maintaining a history of object movement. Object classification is accomplished through a multi-layer, multi-hypothesis approach that does not rely on any static features such as shape or size. Maps are stored in an efficient manner that incorporates a history of previous orientations of each object. GATMO is initialized with a static map; it subsequently changes the map over time as objects in the map change position.

I. INTRODUCTION

Obtaining a detailed understanding of the environment is a key prerequisite to successful robot indoor interaction. In the past few years, simultaneous localization and mapping (SLAM) algorithms have been highly successful at mapping indoor environments, even when the environment is dynamic [17], [12], [5]. Nevertheless, the dynamic and cluttered nature of indoor environments often poses a challenge to robot navigation and localization. There are several ways to react when faced with a dynamic environment. The robot can: A) do nothing, rely on a static map and suffer poor localization; B) ignore its static map for localization and use inertial measurements or scan-matching to localize; or C) create a new map of the environment. Although method A seems naive, it can often suffice in situations where the configuration of the environment has not changed significantly. Method B is much better at dealing with reconfigurations of the environment (if a piece of furniture moved, for example), but it is particularly susceptible to objects that are currently moving, such as people. Method C is generally considered the best solution for dynamic environments, but has one failing: if a new map is created, there is no guarantee that semantic labels of the environment will be preserved. If, for example, an office mail delivery robot was given the task of delivering mail to set destinations (as labeled on a map), making a new map of the environment would necessitate relabeling delivery locations, either through human intervention or some map-matching algorithm. Either way, it would not be practical to generate a new map (through SLAM) every time the room changed. A different strategy, and a fourth option to

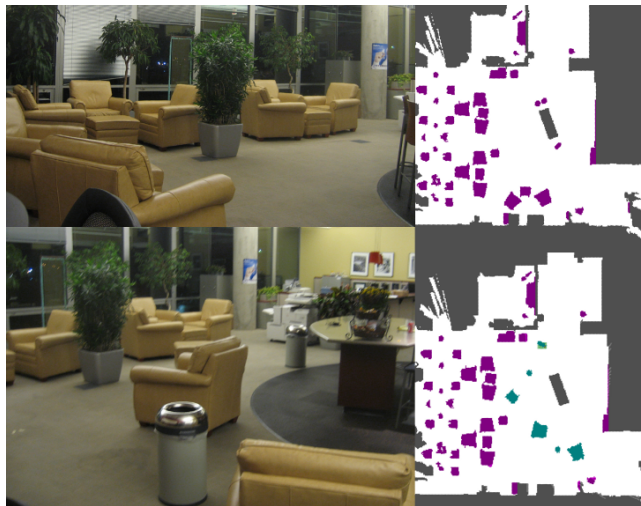interacting in dynamic environments can be found in object representation.

Fig. 1. A situation that requires an object based map. The top image represents the initial configuration of the map. The bottom image is the same map, with the furniture re-configured. By using an object-based map, the robot was able to rearrange the furniture in it's own map to match the real world. The robot accomplished this task by noticing which objects were no longer where the static map indicated, and point-matching those objects to objects it observed. In addition, the robot labeled the moved furniture, as well as furniture in the map that looked similar, as movable.

A much improved method of dealing with a cluttered, dynamic indoor environment is to use an object oriented map that allows for individual object representations. In such a map, all objects in the environment would have their own maps, which can be overlaid to form a static map given the current configuration of the room. This approach has multiple advantages: reconfigurations of the room, which frequently occur, can be easily represented by a simple update of the relevant objects; labeling on objects (such as chair, cabinet) can be maintained, and the robot can even learn the behavior and shape of objects to aid in future recognition tasks. In addition, constructing a complete map of the current configuration would not require observing the entire environment – if the full object shape is known, it will be known in the new configuration of the map, even though the object itself may have only been partially observed. Finally, the knowledge of which objects are movable can assist in navigation strategies that require the robot move objects out of the way.[11] We present such a dynamic mapping system, named GATMO, in which the objects in the map are each individually represented and reconfigurable. In addition, GATMO maintains object data, such as where the

G. Gallagher and J. A. Bagnell are with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA - 15213. S. Srinivasa and D. Ferguson are with Intel Research Pittsburgh, 4720 Forbes Aveune, Suite 410, Pittsburgh, PA - 15213

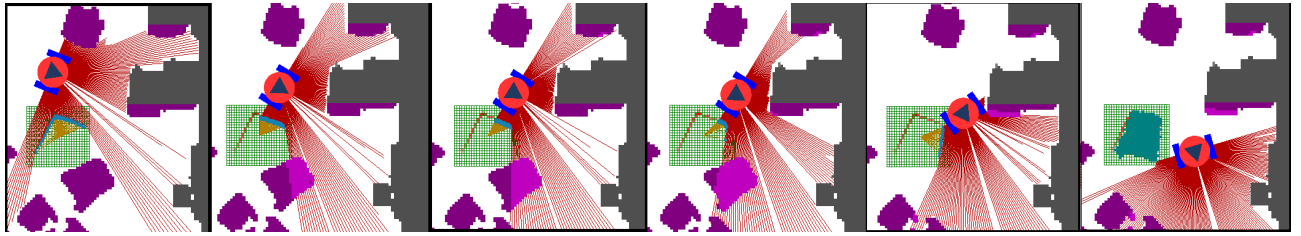*For more information about GATMO, go to www.gatmo.org*

Fig. 2. Recognizing object movement: This figure shows the process of recognizing the absence of an object and re-assigning it to a new location. The robot is shown by the circle with a triangle on it. The red lines extending from the robot show the ray to the laser scan observations that timestep. The purple blobs are objects in the environment, and the grey blob is the wall, colored differently for visualization purposes. The green grid is the grid representation of a *chair* hypothesis. The red areas on the grid are colored according to the log of the probability of that grid cell. The cyan areas of the grid represent the current observations attributed to the grid. The images are as follows *from the left*: in the first image, the robot sees an object, but the set of absent objects, A and the set of movable objects, M, are both empty, so it considers this object to be new to the map. Image two through four: the robot starts to observe that one of the lounge chairs is absent, and begins adding points to the set I_i' corresponding to that object. The points in I_i' are shown in light purple. In image five, the size of the set I_i' for the lounge chair object passes the threshold, and the tuple o_i corresponding to the lounge chair is removed from the set of unknown objects, U, and placed in the set of absent objects A. In image six, the same tuple o_i is compared with the *chair* hypothesis of the metaobject shown, and found to be a match. The pose, p_i of the lounge chair is updated, and the tuple o_i is moved to the set of known movable objects, M. At this time, the object appears in the map as a cyan blob.

object has been seen, and how often it has moved, over many robot interactions. This allows the robot to learn behaviors of individual objects or types of objects. Section 3 will describe how GATMO performs a DATMO (Detection and Tracking of Moving Objects) [5] procedure to locate objects that are not in the map, and to differentiate between objects that are active like people, and objects that are (at least temporarily) static, like furniture. In addition, GATMO performs an object matching technique, described in section 4, using the laser observations that is similar to RANSAC. (See figure 3 for the full GATMO structure.) Although the concept of offline mapping with object representations has been presented in the past [1] [7] , GATMO stands out in its ability to identify dynamic objects in the environment in an online manner.

II. RELATED WORK

This paper builds on many areas of research, including people tracking, dynamic object detection, object recognition, dynamic mapping, and point matching. The contribution of GATMO is to merge the areas of people detection and dynamic object detection with the area of object recognition and dynamic mapping to allow the latter to be done in an informed an online manner.

There have been a number of works involving detecting people and other dynamic objects using laser data. Montemerlo presents SLAPT, Simultaneous Localization And People Tracking using conditional particle filters [12]. Nunez and Mendes describe the Detection and Tracking of Moving Objects (DATMO) in their 2004 paper[14]; the DATMO algorithm works by removing observations of static objects, clustering the remaining observations, assigning them to objects based on a number of features, and tracking them using Kalman filters. Wang combines DATMO with SLAM using Kalman filters to calculate joint distributions over robot position and object pose [13]. Haenel also presents a method of map building in dynamic environments that does not encode specific features of the dynamic objects in order to detect them. [18] Shultz and Burgard use a different method of people tracking, which focuses on the data association

problems, using Sample-based Joint Probabilistic Data Association Filters (SJPDAF), [15].

GATMO's approach to dynamic object detection is built off of the DATMO model, but unlike Mendes, Nunes and Wang, it uses no pre-programmed features to classify objects. Instead, it uses point matching to directly match current observations to previously seen objects. In addition, our work separates itself in its intention to store data about dynamic objects, and its focus on objects that occasionally move, rather than filtering out transitory objects.

In perhaps the most closely related work in the field of dynamic object detection, Shultz and Burgard present a method of determining, based on laser readings, when an object that was previously mapped has changed position. [16]. In this work, Baysian state estimation is used to estimate both the robot's position and the objects' orientation. While this paper successfully identified object poses, it assumed that objects remain in approximately the same area. GATMO builds on the work of Schulz and Burgard by accounting for changes in position and orientation.

When compared with detecting dynamic objects, the field of semantic mapping is still fairly young. Current approaches to semantic labeling mostly rely on static features of the environment, rather than dynamic properties of objects. For example, Rottmann, et. al.[7] describe a supervised learning approach to optain the label of a room based on it's visual and laser features. Only very recently has work been done to apply labels based on dynamic data; in [21] Wolf and Gaurav explore activity-based mapping in an outdoors setting.

Finally, some work has been done in the area of object recognition with laser scanners. [4] uses an EM based approach to perform offline object recognition by comparing multiple static maps. An online method based on angular constraints is presented in [2], but this is unable to distinguish objects with the same convex hull.

III. MAP STRUCTURE

The map used in GATMO consists of two main parts, a static map, and several lists of objects with locations and

orientations on the map. The lists of objects are given in

List Name	Object Description	Comment
A	Absent	Was in the map, but is now absent
M	Movable	Has moved, currently in the map
U	Unclassified	Default object state
Γ	Never Added	Observed, but never added

TABLE I

LISTS MAINTAINED IN A GATMO MAP

Table I.

To begin, we define a map space G, that for simplicity will be gridded into square grids with sides of length δ_R. Each grid square x_i has an 8 connectivity with the points immediately adjacent to it, expressed as $neighbors(x_i)$. GATMO is initialized with a static map, expressible as the set of all points S_0 that are occupied, and S_u, the points which are not. GATMO then performs region growing to obtain sets of contiguous occupancy I_i. Map objects are represented as $o_i = \{(I_i, E_i, N_i, p_i)\}$, where I_i is a list of all points in the set, E_i is the set of points on the edge of I_i, N_i is the set of points in G that are closer than δ_W to some edge point, and p_i, the position of the set.

As a map, GATMO maintains three sets, U, M, and A, (see Table I). All tuples o_i are initialized to be in U, since they are unclassified. As the robot notices changes in its environment, it reclassifies objects, moving them from U to A or M.

For each tuple o_i, a local grid is maintained which includes all points in I_i, E_i, and N_i, and is related to the global frame by p_i. This way, if the object represented by o_i moves, only p_i needs to be updated.

A. Object Reclassification

As the robot roams the environment, it either sees the objects in the positions that it had recorded, or it sees through their location if the object is absent. If the object is absent, or has moved, the map records when a laser passes through where it believed the object to be by marking the cells on the object's grid representation. Sets of observations $z_i \in Z_t$ are taken from the laser. Let $Ray(z_i)$ represent the set of points in G on the line between the laser scanner's pose and the observation point. We maintain a set I'_i for every $o_i \in U, B$, and perform the following procedure:

Algorithm 1 Determine if an object is still in the map by ray-tracing laser data

  for all $z_i \in Z_t$ **do**
    $I'_i = I'_i \cup (I_i \cap Ray(z_i))$
    if $size(I'_i) > \beta_{thresh} \times size(I_i)$ **then**
      Move o_i to A
    end if
  end for

If the object's 'unseen' cells, I'_i is over a percentage threshold β_{thresh}, the object is added to the absent list, and

no longer shows up on the map. A possible expansion of Algorithm 1 would be to have I'_i keep track of a probability rather than a binary value. However, in our implementation, the number of false deletions was negligible, so this extension was unnecessary.

Once on the absent list, the object can be re-classified as movable if it is observed in another location (see section V.) When the robot uses the map for future runs, the classifications remain, with the exception that absent objects are not included in the static map. Movable objects retain a high prior of being seen their most recent locations, and both absent and movable objects maintain a prior of being seen in any of their previous locations.

IV. PEOPLE TRACKING AND OBJECT CLASSIFICATION

In this section, we introduce our multi-hypothesis object classification hierarchy, and then describe how it is used to classify laser data.

A. Object Classification Hierarchy

GATMO represents each object by a *metaobject* which maintains multiple hypotheses H of what it is. We define two hypotheses, $H = \{h_{person}, h_{chair}\}$. The $person$ hypothesis represents objects that actively move, like people, animals, and other robots. The $person$ hypothesis maintains a Kalman filter that tracks the position and velocity of the object's center. The $chair$ hypothesis represents objects that seldom move. (Note that this does not mean that the object *is* a chair, but rather a seldom-moving object.) It is represented by a dynamic grid, which encodes past observations. However, instead of recording observations like an occupancy grid, the $chair$ hypothesis simply adds probability where observations have been made, and does not subtract probability from cells between the observation and the robot. This strategy enables the $chair$ hypothesis to model objects that an occupancy grid might miss. The $chair$ hypothesis also maintains several association hypotheses. In addition to considering itself as a new object on the map, the $chair$ hypothesis considers the probability that it is an object in A, the absent list of map objects, M, the movable list of map objects, and Γ, a set of previously seen grid objects that have not been added to the map. At any particular point in the robot's operation, a set of metaobjects are maintained which represent objects seen at the current timestep, or in the past τ_{memory} seconds. When a *metaobject* is discarded after not being seen for τ_{memory} seconds, it checks the dominant hypothesis. If the $chair$ hypothesis was dominant and the $chair$'s dominant association hypothesis was not to a map object, the $chair$ hypothesis is added to Γ. Nothing is done if the $person$ hypothesis is discarded, or if the $chair$'s dominant association hypothesis was to a map object.

B. Object Classification Procedure

The object identification and classification procedure involves multiple stages. First, the robot pose is estimated, and the laser scans are aligned to the environment. GATMO uses AMCL to update global position, and perform scan matching
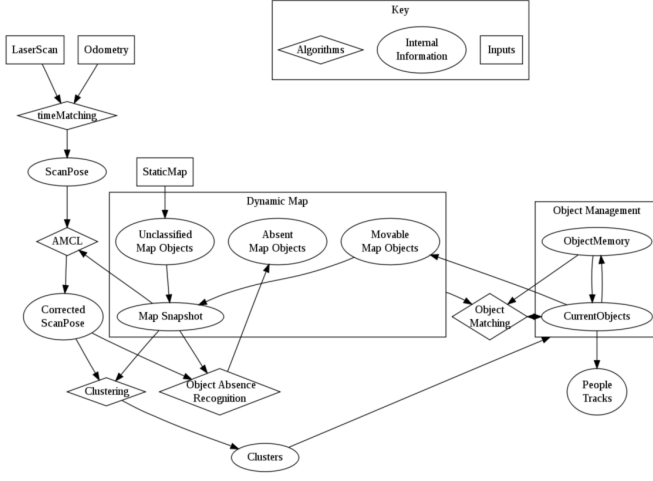
Fig. 3.   Structure of the GATMO system

between global updates for a precise position estimate. We use the covariance of the AMCL's particle filter as an indicator of when to reset the scan matching offset. After the robot has been aligned to the global frame, GATMO performs the following object classification procedure, outlined in Figure 3:

We define a set of points $S_{free} \subset S_u$ that are unoccupied and not close to any unclassified object. Formally, $S_{free} = \{x_i \in G | \forall N_i \in U, x_i \notin N_i\}$ Note that neither S_{free} nor S_u is ever explicitly calculated, and S_0 is only calculated once when the robot is first initialized.

Sets of observations Z_t are taken from the laser. Next, the observations that correspond to the unclassified objects are filtered out by comparing them with S_{free}:

$$Z_{free} = S_{free} \cap Z_t$$

Using Mean Shift, we segment the observations into N disjoined sets of observations:

$$Z_{clust} = \{\{z_1, z_2, ..., z_N\} \mid \bigcup_{i=1}^{N} z_i = Z_{free}\}$$

We then form a set of clusters, $C = \{c_1, c_2, ..., c_N\}$ which we use to associate the observations with our objects Ob_{t-1}:

$$c_i = \{(z_i, \alpha_i) \mid \alpha_i = \underset{\{m_j \in Ob_{t-1}, m_{new}\}}{\operatorname{argmax}} P(z_i | m_j)\}$$

where m_j is a metaobject, having in general n hypotheses h_k. m_{new} represents creating a new object for the cluster. The probability of a cluster being associated with the metaobject is as follows:

$$P(z_i | m_j) = \sum_{h_i \in H} P(z_i | h_i)$$

$$P(z_i | h_{person}) = \prod_{k=1}^{K} P(x_k \in z_i | p_j)$$

$$P(x_k | p_j) = \int_{x' = x_k^0 - \delta_R}^{x_k^0 + \delta_R} \int_{y' = x_k^1 - \delta_R}^{x_k^1 + \delta_R} f_{X,Y}(x', y') dx' dy'$$

where p_j is the person hypothesis of the object m_j, and $f_{X,Y}(x', y')$ is the pdf of the multivariate normal distribution $N(\mu_j, \Sigma_j)$ maintained by the Kalman filter associated with p_j.

The chair hypothesis is as follows:

$$P(z_i | h_{chair}) = \prod_{k=1}^{K} P(x_k \in z_i | r_j)$$

The probability $P(x_k \in z_i | r_j)$ is also taken from a multivariate gaussian distribution around all previous observations. This value can be obtained by querying the value of the gridcell on which the laser observation landed. An approximation is used if x_k falls outside the grid maintained by the chair hypothesis, but such values are very low, and do not have a significant impact on the algorithm as a whole.

The metaobject m_{new} always returns a probability of β_{min_prob}, representing the minimum probability $P(z_i | m_j)$ that can be returned and still associate with the cluster c_i. Below this threshold, c_i is considered a new object in the scene.

After all the clusters are assigned, an update step is performed, when all the observations are added to the appropriate metaobjects. The clusters are added to both hypotheses, regardless of whether the hypothesis is dominant. The Kalman filter of the $person$ hypothesis is updated, and the grid associated with the $chair$ hypothesis is updated, as described in the previous section.

Finally, the dominant $chair$ hypotheses are compared to the sets in A and M. If the chair grid matches one of these objects, the map object is classified as movable and repositioned accordingly. The details of the object matching are covered in section V. Once a tuple o_i is classified as movable, the set is then compared against all the other unclassified sets (those in set U). If the tuple o_i matches any in U, those sets also are classifieds movable. It should be noted that the observation times of the $chair$ hypotheses and the sets in A do not influence the object matching procedure. This means that an newly observed object can be matched to the absence of an object elsewhere, regardless of which was observed first.

At the end of a run, or if the robot has observed over a set percentage of the map (representing total coverage), GATMO adds the dominant $chair$ hypothesis that were stored during the run to the map as map objects. To convert the grid structure to a map object, the cells with a value over a set threshold are added as points in the map. The grid is added as a set S in the set of movable objects, M. In this way, new objects can be added to the map after initialization.

V. OBJECT MATCHING

In order to maintain semantic labels for objects as they move, the observations must be matched to the map objects, even though they have undergone translations and rotations. Several object matching techniques have been previously
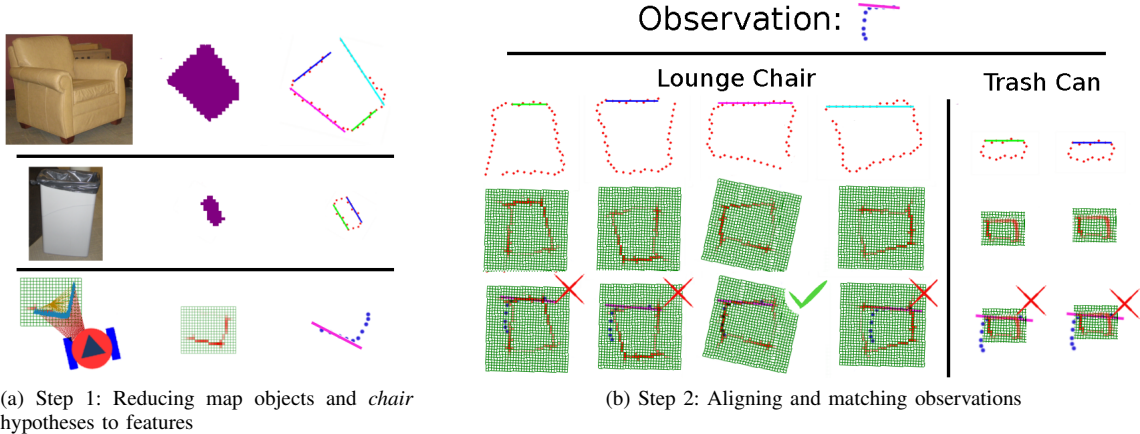
(a) Step 1: Reducing map objects and *chair* hypotheses to features

(b) Step 2: Aligning and matching observations

Fig. 4. **(a)** The columns in this figure show, from left to right: The object being observed, either originally in the map, or when the robot is roaming the environment; the object's representation in the map, as a blob of points for map objects, or a grid for *chair* objects; and the extracted edges of the object with lines fitted to them. **(b)** The objects from figure 4a are aligned. The aligned observation, at the top, is then compared to grids formed by adding a Gaussian distribution around each edge point in the set E_i and discretizing. Both the observation and the set of edge points of the map object are translated so as to have a mean of zero. A simple search is performed in the grid space around the initial alignment, in order to find the best match.

| | REAL OBJECT | | |
	Footstool	Trash Can	Chair
MATCHED OBJECT			
Footstool	**806**	42	92
Trash Can	47	**478**	0
Chair	236	150	**1748**

TABLE II: ACCURACY OF OBJECT MATCHING.

proposed, including an EM based method [4], performed in an offline setting, and an online method based on angular constraints [2]. The method used in GATMO is more simplistic, but still very effective. When considering an object, the edges are extracted and lines are fit to the sets of edge points (See figure 4a.) The principal axes of the points that made up the line are found using SVD, and the object is rotated to lie on these principal axes.

As multiple lines usually exist on the object, multiple transforms (usually between two and five) are found. Thus multiple objects, similarly transformed, can be compared by checking the distance between corresponding points in the two object. In other words, for two objects, O_1 and O_2, with the sets of points x_1^i and x_2^j, the location of each point x_1^i is compared to the location of the closest points in x_1^j. This matching is done in the same manner as the cluster observations to *chair* hypotheses in section IV. By placing one object on a grid, the entire matching can be done in $O(nk)$ time, where n is the number of edge points and k is the number of object templates available for assignment.

VI. EXPERIMENTAL SETUP

The data for our experiments was collected using a SICK LMS 200 mounted horizontally 12 cm off the ground on a segway rms platform. The robot used the Player 2.0.4 architecture [9], including the amcl driver provided for localization. Scan matching was performed by using the scan matching library written by Dirk Haenel for Carmen. Object matching was performed using the SVD functions provided

Object	Chair	Rect. Can	Circ. Can	Box
Time (seconds)	3.50	0.90	0.75	1.50
Laser Scans	70	15	10	30

TABLE III: AVERAGE TIME TO CLASSIFY

in the Gnu Scientific Library (GSL)[10]. Experiments were performed in the Intel Lab in the Collaborative Innovation Center at Carnegie Mellon. Data was taken over a series of runs that spanned several weeks. We tested several aspects of GATMO, including matching accuracy (choosing the correct object and orientation), positional accuracy of the match, and number of observations necessary to make a correct match. We performed the tests by navigating the robot around the room for 25 runs, with runtime ranging from three to ten minutes per run. Between each run, we moved between two and seven objects. In Table II, a confusion matrix is shown for three objects that were moved and identified in most of the runs. Listed in Table II are the number of scans that supported each match. Figure 5 shows the percentage of correct matches made as a function of number of scans. Smaller objects tended to be harder to classify uniquely, but they could be classified sooner as apposed to larger objects (like the lounge chair) which took on average 70 scans before it could be correctly identified.

VII. CONCLUSIONS AND FUTURE WORKS

We presented an online approach to object matching and object oriented mapping. We evaluated the matching performance for a variety of objects in many different poses. Overall, GATMO performs very well. It was able to detect objects that were not in the map, detect new objects in the environment and match old objects to new objects. The entire process is done in an online fashion, and the resulting map improves both localization and navigation. In addition, the labeling of the object is maintained when the map is changed, which will allow the development of many future applications involving specific objects in the room.
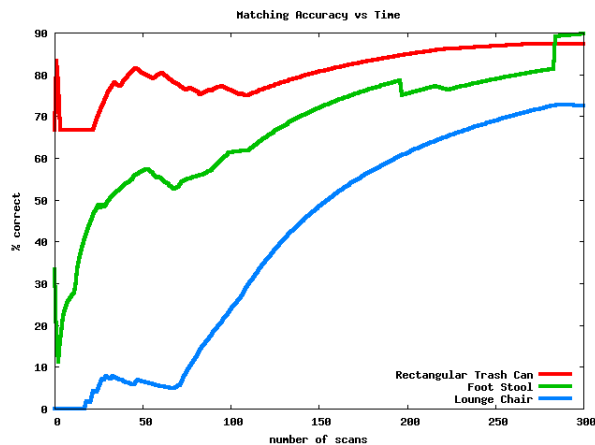
Fig. 5.   Object identification versus time

This paper opens the possibilities for many new directions in object oriented interaction with the environment. For example, a robot could track the behavior of objects over time, to identify patterns (such as, an extra chair is moved into the dining room every Sunday) that it can act upon. The robot could plan navigation trajectories using probable configurations of the environment (the route through hall 1 is blocked more often than hall 2). Work has already been done in the area of navigating in highly cluttered environments by moving objects out of the way [11]; this work could be expanded by changing the binary mobility indicator (in the set U vs set M) in this paper to include more information on how objects can be moved. Finally, the robot could even 'tidy' the room by putting objects back where they were first observed.

There are a number of improvements to GATMO that will be explored in future work to make it more robust. While object matching was very reliable for larger objects, smaller objects tended to be confused with each other. In addition, larger objects tended to take a larger time to match. Although we feel that the simple approach to object matching we describe in this paper was sufficient to demonstrate the concept of online object oriented mapping, we feel that GATMO could benefit from a more intelligent approach to point matching.[19][20] Another important factor that was not fully explored in this paper is the subject of localization. While the map that AMCL is using is updated when an object changes position, GATMO currently does not take special advantage of this new knowledge. It would be useful to integrate the map updates, as well as other object detections more directly into the localization algorithm.

Another useful addition to GATMO would be to integrate camera data into the object matching and people tracking. By using camera data, the time to associate objects and determine the dominant hypothesis would be reduced, and the object matching accuracy would be greatly improved.

VIII. ACKNOWLEDGMENTS

REFERENCES

[1] D. Anguelov, R. Biswas, D. Koller, B. Limketkai, S. Sanner, and S. Thrun, Learning hierarchical object maps of non-stationary environments with mobile robots, in Proc. of the Conf. on Uncertainty in Artificial Intelligence (UAI), 2002.
[2] J. Modayil and B. Kuipers, "Autonomous Shape Model Learning for Object Localization and Recognition," Proc. of the Int. Conf on Robotics and Automation, May 2006
[3] J. Modayil and B. Kuipers, Bootstrap learning for object discovery, IEEE/RSJ Int. Conf on Int. Robots and Sys., 2004, pp. 742
[4] R. Biswas, B. Limketkai, S. Sanner, and S. Thrun, Towards object mapping in non-stationary environments with mobile robots, in IEEE/RSJ Int. Conf on Int. Robots and Sys., vol. 1, 2002
[5] C.-C. Wang, C. Thorpe, and S. Thrun, Online simultaneous localization and mapping with detection and tracking of moving objects: Theory and results from a ground vehicle in crowded urban areas, in IEEE Int. Conf. on Robotics and Automation, 2003, pp. 842 849.
[6] B. Limketkai, L. Liao, and D. Fox, Relational object maps for mobile robots, in Proc. of the Int. Conf. on Artificial Intelligence (IJCAI), Edinburgh, Scotland, 2005, pp. 14711476.
[7] O. Mozos , A. Rottmann, R. Triebel , P. Jensfelt, and W. Burgard, "Semantic Labeling of Places using Information Extracted from Laser and Vision Sensor Data" in IEEE/RSJ Int. Conf on Intelligent Robots and Systems, 2006
[8] S. Thrun, D. Fox, W. Burgard, and F. Dellaert. "Robust Monte Carlo Localization for Mobile Robots." Artificial Intelligence (AIJ), 2001.
[9] Brian P. Gerkey, Richard T. Vaughan, Kasper Stoy, Andrew Howard, Gaurav S. Sukhatme, and Maja J Mataric. "Most Valuable Player: A Robot Device Server for Distributed Control". In Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS 2001), pages 1226-1231, Oct 2001.
[10] "GSL - GNU Scientific Library", http://www.gnu.org/software/gsl/ Last Accessed September 15, 2008
[11] M. Stilman, "Navigation Among Movable Obstacles" doctoral dissertation, tech. report CMU-RI-TR-07-37, Robotics Institute, Carnegie Mellon University, October, 2007
[12] Montemerlo, M., Thrun, S.: Conditional particle filters for simultaneous mobile robot localization and people-tracking (slap). In: ICRA. Proc. of the IEEE Int. Conf.e on Robotics and Automation (2002)
[13] Wang, C.-C., Thorpe, C.E.: Simultaneous Localization And Mapping With Detection And Tracking of Moving Objects. In: ICRA. Proc. of the IEEE Int. Conf. on Robotics and Automation (2002)
[14] Mendes, A. Bento, L.C. Nunes, U. "Multi-target detection and tracking with a laser scanner", Intelligent Vehicles Symposium, June 2004
[15] D. Schulz, W. Burgard, D. Fox, and A.B. Cremers, "People Tracking with a Mobile Robot Using Sample-based Joint Probabilistic Data Association Filters." International Journal of Robotics Research (IJRR), 22 (2), pp. 99-116, 2003.
[16] D. Schulz and W. Burgard, "Probabilistic State Estimation of Dynamic Objects with a Moving Mobile Robot.", Robotics and Autonomous Systems, 34 (2-3), pp. 107-115, 2001.
[17] C. Stachniss and W. Burgard, "Mobile Robot Mapping and Localization in Non-Static Environments." In Proc. of the National Conference on Artificial Intelligence (AAAI), 2005.
[18] D. Hhnel, R. Triebel, W. Burgard, S. Thrun. "Map Building with Mobile Robots in Dynamic Environments." In Proc. of the IEEE International Conference on Robotics and Automation (ICRA), 2003.
[19] S. Gold, A. Rangarajan, C. Lu, E. Mjolsness," New algorithms for 2d and 3d point matching: Pose estimation and correspondence." Pattern Recognition, 31(8), 1998
[20] S. Belongie, J. Malik, J. Puzicha, "Shape matching and object recognition using shape contexts", IEEE Transactions on Pattern Analysis and Machine Intelligence,2002
[21] Denis F. Wolf and Gaurav S. Sukhatme, "Semantic Mapping Using Mobile Robots", IEEE Transactions on Robotics, VOL. 24, NO. 2, April 2008