



Tracking and Following Algorithms for Mobile Robots for Service Activities in Dynamic Environments

Feng-Li Lian^{1,*}, Chin-Lung Chen¹, and Chih-Chung Chou¹

¹Department of Electrical Engineering, National Taiwan University

(Received 31 July 2014; Accepted 12 November 2014; Published on line 1 March 2015)

*Corresponding author email: fengli@ntu.edu.tw

DOI: [10.5875/ausmt.v5i1.838](https://doi.org/10.5875/ausmt.v5i1.838)

Abstract: Providing robots with the capability of following human targets can allow them to assist people in various ways in different environments. One of the main difficulties in performing human tracking and following is the occlusion problem caused by static and dynamic obstacles. This paper addresses the occlusion problem by planning a robotic trajectory to maximize target visibility and following the moving target. Initially, a laser range finder is used to detect the human target and then robustly track the target using the Kalman filter. Afterward, a human following algorithm based on a look-ahead algorithm, DWA*, is implemented to pursue the target while avoiding static or dynamic obstacles. Extensive experiments were conducted to test to evaluate robot maneuvers and several field tests were conducted in more complex environments such as a student cafeteria, computer center, and university library.

Keywords: Human following, path planning, dynamic window approach, detecting and tracking moving object

Introduction

One fundamental skill of a robot is moving from its present location to a specified destination. Several studies have been devoted to topics including localization, mapping, and path planning for this purpose. Other studies have focused on the use of different sensors such as laser range finder and/or camera to detect and/or track moving objects. This paper seeks to improve the robot's ability to follow a human target by further integrating algorithms used in moving object tracking and path planning.

The ability of robots to track pedestrians in their surroundings is essential to many real life applications such as providing assistance in venues such as museums, offices or libraries. Another key aspect of human-robot interaction is the robot's ability to follow a target pedestrian in an indoor environment while executing various instructions such as holding books in a library or carrying groceries at a store. Figure 1 illustrates a schematic scenario of a robot following a target pedestrian. The robot should follow the target while avoiding a static obstacle (indicated by the red triangle), and a moving obstacle (the secondary pedestrian). This

paper seeks to address this type of problem by allowing the robot is to accurately identify the target and robustly follow the target while avoiding any obstacles.

The main tasks for each stage are listed as follows. As shown in Fig. 1(a), the robot executes both 'Localization' and 'Moving Target Tracking' operations. That is, the robot must first localize itself in the environment and then identify and track the target human. Once the robot starts to follow the target as shown in Figs. 1(b) and 1(c), the 'Target Tracking' process provides the estimated target location and trajectory while the 'Path Planning' generates and selects the optimal path to follow the target, as shown in Fig. 1(d). After the path is selected, the velocity commands are given to the 'Actuator' to initiate the movement. This paper focuses on the integration of 'Localization', 'Moving Target Tracking', and 'Path Planning' to achieve human following.

A robust approach to follow a target human in dynamic environments is proposed. Several techniques have been used to improve robot localization and human tracking results. Moreover, the detecting and tracking moving object (DATMO) system is integrated with the dynamic window approach (DWA*) [16] navigation system to anticipate human behavior and, as a result, to enhance human following performance. The robot is capable of



selecting the optimal path by simultaneously considering several operational factors such as the distance between the robot and the target, and the target visibility at each moment. This ensures that the robot follows the target in a robust and appropriate manner.

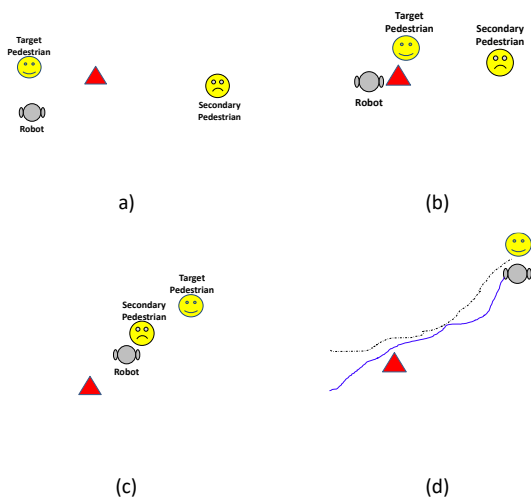


Figure 1. Pedestrian following scenario.

The remainder of this paper is organized as follows. Section 2 discusses some of the important related works. Section 3 describes the proposed DATMO system in detail, including the methods used in robot localization, moving point detection, and human tracking. Section 4 presents the framework for integrating the DATMO and DWA* approaches. Section 5 compares the proposed algorithms and evaluates the related experimental performance. This section also describes the experimental results of the robot following a human target in real-life indoor

environments. Finally, Section 6 concludes this paper and provides important directions for future work.

Background and Literature Survey

Various approaches have been used to detect and track moving objects using a laser range finder. When detecting moving objects in the robot's surroundings, Wolf & Sukhatme [1] proposed the concept of maintaining static and dynamic occupancy grid maps. Landmark features are used to solve the localization problem when the robot is in motion. A different approach, presented by Horiuchi et al. [2], searches for local minima to recognize humans and uses the particle filter localization method. However, this technique cannot resolve the issue of frequent occlusions. Yang and Wang [3] proposed a robust ego-motion estimation algorithm in dynamic environments.

For tracking moving objects, Schulz et al. [4] and Mucientes et al. [5] proposed using probabilistic methods such as multiple hypothesis tracking (MHT), joint probabilistic data association (JPDA), and Kalman filter to develop a human tracking system. On the other hand, Shao et al. [6] introduced the idea of combining a human walking model with a particle filter to track hundreds of people with stationary laser range finders. Lee et al. [7] applied a similar walking model to a robot in motion using odometry data, and their experimental results showed accurate human tracking assuming perfect odometry.

This paper aims to solve the moving target following problem in the face of obstacles. Many previous works have successfully demonstrated automatic driving in static environments, including Fox et al. [8], Ulrich and Borenstein [9], Minguez and Montano [10] and Seder and Petrović [11]. However, these methods are designed to reach a fixed goal and assume that the environment and robot states are fully observable. Directly applying traditional obstacle avoidance algorithms on the target following task directly can fail easily because a moving target can change its speed and direction at any time and the target can be occluded by obstacles.

The partially observable Markov decision process (POMDP) provides a general framework for planning under imperfect perception. However, using POMDP to compute the optimal policy is usually very computationally expensive because it has to compute a plan over a large belief space (typically $N-1$ dimensional for an N -state problem). Fortunately, several useful approaches for POMDP approximation or alternatives have been developed in recent years. Many aim to reduce the dimension of belief space, such as PBVI in Pineau et al. [12], AMDP in Roy et al. [13], and MOMDP in Sylvie et al. [14]. These methods are much faster than the original

Feng-Li Lian received the B.S. and M.S. degrees from National Taiwan University in 1992 and 1994, respectively, and the Ph.D. degree from the University of Michigan in 2001. From 2001 to 2002, he was a postdoctoral scholar at California Institute of Technology. Since 2002 he has been in the Department of Electrical Engineering, NTU, and, from 2009 to 2013, he was also the Division Director of Information Management, Computer & Information Networking Center, NTU. He is the recipient of the Youth Automatic Control Engineering Award from Chinese Automatic Control Society, Taiwan, in 2007, the Outstanding Youth Award from Taiwan Association of System Science and Engineering in 2012, the Dr. Wu, Da-You Memorial Research Award, National Science Council, Taiwan, in 2012, the Excellent Young Scholar Research Grant, National Science Council, Taiwan, in 2012-14, and the NTU Excellent Teaching Award in 2007, 2008, 2010, 2011, 2012 and 2013. His current research interests include distributed and networked control systems, multiple dynamical agent systems, trajectory generation and path planning.

Chin-Lung Chen received the B.S. degree from Ryerson University, Canada, and the M.S. degree from National Taiwan University, Taiwan. Currently, he works research and development engineering in in robotics-related industry.

Chih-Chung Chou was born in Taipei, Taiwan, in 1985. He received the B.S.E.E. and M.S.E.E degrees from National Taiwan University, Taipei, in 2007 and 2009, respectively. He is currently Ph.D. candidate in the Department of Computer Science and Information Engineering, National Taiwan University. His research interests include robotics and artificial intelligence.



POMDP, but are still too computationally complex to perform real-time planning. The method proposed here is a sub-optimal but fast approach which assumes that the robot always receives the most possible observation and plans a path which can reach the goal and minimize uncertainty in particular states. For example, in Prentice and Roy [15], the robot aims to reach the goal with minimum uncertainty for its position, so it may choose a path which is long but provides sufficient localization landmarks.

We propose a motion planner for following a moving target. The robot is first effectively localized by the scan matching method and several techniques are utilized to filter out outliers and other moving objects. We then combine the concepts of the occupancy grid map and feature recognition such as local minima and motion velocity to improve human detection accuracy. Each human is then tracked by using the extended Kalman filter (EKF) which solves the occlusion problem and also provides the computational advantage in real time performance. Finally, the planner uses an extension of the dynamic window approach proposed in Chou et al. [16] and Chou and Lian [17] to find collision-free velocities and choose a proper velocity using the A* heuristic search [22]. Proper cost functions are designed to minimize the distance between the robot and the target and to maximize the possibility that the robot can maintain observation of the target in a fixed time horizon. Similar to the work of Pomares et al. [18] which uses a collision avoidance system in human-robot cooperation, the proposed path planner can also guarantee human safety and collision avoidance. In addition, the concept of the nearness diagram algorithm proposed by Minguez and Montano [10] is applied to compute a better estimation of the distance between the robot and target, thus achieving a smooth, non-hesitating performance.

Detecting and Tracking Moving Objects

Detecting and tracking moving objects (DATMO) is a long-studied issue in mobile robotics. This paper aims to solve the DATMO problem using a laser range finder mounted on a mobile robot. This approach is much more difficult than implementing DATMO using a fixed sensor because the robot must localize itself correctly in order to distinguish static objects and moving objects. The proposed system architecture refers those studied in Wang et al. [19] and Montesano et al. [20]. First, robot odometry is corrected by a scan-matching algorithm with outlier filtering. Second, moving objects are detected by comparing the current scan with a local occupancy grid map. Finally, for each moving object, the Kalman filter is used to maintain object tracking.

Accurate odometry is a key for detecting moving objects using a mobile robot. Most studies on moving target tracking assume that the robot has perfect odometry, but this is a poor assumption in most real environments. The iterative closest point (ICP) algorithm is a popular method for correcting robot odometry by finding the relative position between two laser scans. However, the standard ICP algorithm assumes that all points are static and contained in these two compared scans. In real applications, outlier points such as moving objects or occluded objects may seriously influence ICP performance. Hence, outlier deletion is a very important issue when using ICP odometry.

In the standard ICP algorithm, each laser scan is matched with the previous scan to acquire the robot pose. However, this method incurs accumulated errors from each scan-matching. In other words, once the robot pose is inaccurately computed, the scan-matching result will suffer the same inaccuracy since only the previous scan is used. To remedy this problem, N (e.g., $N=5$) previous scans are also used to perform scan-matching with the current scan, namely, multiple ICP or MICP, instead of only the previous one. Thus, even if the previous pose was inaccurately computed, the ICP algorithm still has plenty of accurate scan points to match with and acquire the correct pose.

When performing scan matching in the dynamic environment, moving points can significantly degrade scan matching performance which depends on the number of moving objects. It is imperative that these moving points are eliminated prior to scan-matching and only static points from the environment are used. When a new measurement scan is acquired in the system, it consists of both stationary and moving points. It is difficult to accurately separate the stationary and moving objects at this point. Therefore, the robot pose generated from MICP using this measurement scan will lose some accuracy. Nonetheless, this pose is required to detect the moving objects along with the global static map. Once the moving objects are identified, MICP can be performed again using only the stationary points this time. Hence, the accuracy of newly acquired robot pose improves. Finally, the detecting moving object algorithm is performed again using the most accurate robot pose.

Furthermore, the occupancy grid map is used to segregating moving objects in dynamic environments. In [1], Wolf and Sukhatme proposed the idea of maintaining two grid maps: static and dynamic. In this paper, a static map is generated from the past N frames. This static map is updated when a new measurement scan is acquired. A grid cell can be in one of three states: Free (probability < 0.2), Unknown ($0.2 < \text{probability} < 0.8$), and Occupied (probability > 0.8). The inverse observation model



determines the updated rate of the cell probabilities or how fast a grid cell shifts from the Free state to the Occupied state and vice versa. Table 1 depicts the detail of inverse observation model used in the system.

Table 1. Inverse observation model.

S^{t-1}	o^t	$p(S^t o^t, S^{t-1})$
Free	Free	Low (0.2)
Unknown	Free	Low (0.3)
Occupied	Free	Low (0.2)
Free	Occupied	Low (0.3)
Unknown	Occupied	High (0.7)
Occupied	Occupied	High (0.8)

Note that S^t denotes the state at time t , o^t denotes the observation, i.e., sensory measurement, at time t , and $p(S^t|o^t, S^{t-1})$ denotes the conditional probability under o^t and S^{t-1} .

The sensor data is first clustered into numerous objects. The human detection process segregates the data from the humans and the noise from the batch of clusters, where each human is tracked independently in tracking human process. Ultimately, the human data are stored in the human database and are constantly updated. Each process block is discussed in the following section.

Upon acquiring all the moving points in the laser scan, these data points can now be grouped into various clusters. Each cluster represents a moving object, human or otherwise. This paper uses the point-distance segmentation method which computes the Euclidean distance between two consecutive scan points, i.e., P_i, P_{i+1} , in the polar coordinate, using the following formula:

$$D(p, p_{i+1}) = \sqrt{r_{i+1}^2 + r_i^2 - 2r_{i+1}r_i \cos(\theta_{i+1} - \theta_i)} \quad (1)$$

where r_i and θ_i are respectively the measured distance and the indexed angle of the i -th data from the laser range finder. If the calculated distance given by Eq. (1) is less than a threshold value, D_{th} , the points are considered to be belong to the same object. Dietmayer et al. [21] proposed a threshold function to calculate D_{th} which allows the adjustment to noise and overlapping in close range. The D_{th} value used in this paper is set at 0.1 m, which has been experimentally shown to be an adequate clustering range. Each cluster is then represented by its center-of-gravity (COG) and available for the next step of the tracking procedure.

The average width of a human leg is approximately 0.1 m to 0.3 m. Consequently, to become a valid leg candidate, the cluster size must be within the range of 0.1 m and 0.5 m. The upper limit is set at 0.5 m because, more

often or not, two legs are grouped into one cluster due to occlusion depending on the motion of the human. Furthermore, any clusters with fewer than five scan points are automatically eliminated.

One feature that can be observed and used to separate humans from noise is the moving velocity. It is assumed that a human moves at a velocity greater than a threshold value λ while the noise should have very minimal or no velocity. In other words, the clusters must move at a sufficient velocity to qualify as leg candidates. The λ used in this paper is 0.5 m/s.

On the other hand, most noise is caused by accumulated errors in SLAM and LRF inaccuracy. These falsely identified moving points are essentially static points from obstacles or walls in the environment. As a result, the local minima of the moving clusters belonging to a human are distinguishable to one from a noise cluster. Thus, a valid leg candidate must have a local minimum greater than a predefined threshold value γ . Consequently, any falsely identified moving clusters from the static environment can now be eliminated.

When moving clusters are initially detected, they are first assigned as dummy tracks. Each of these clusters is given three-scan periods to become a valid leg candidate by satisfying the following three filtering conditions:

1. 0.1 m < cluster size < 0.5 m;
2. Moving velocity > λ m/s;
3. Local minimum > γ m.

Once a cluster turns into a leg candidate, it searches for the other leg candidate in the proximity of 0.5 m. If no other leg candidate is found, the sole leg candidate is assigned a unique ID and represents a human. On the other hand, if two leg candidates are found within the range, they are considered to belong to the same human and assigned a single ID. If the observation indicates that the cluster does not fit the characteristics of a human, it is considered noise and removed from the database.

One of the main objectives in target tracking is to estimate the motion parameters of the moving target. The extended Kalman filter (EKF) is a computationally efficient estimator for the state of a dynamic system observed through noisy measurement. The noise processes are assumed to be Gaussian and the motion behaviors are assumed to be linear. In this work, the state vector used in EKF is $X = [x, y, v_x, v_y]$, representing the position and velocity of the human in the x and y directions. Each iteration involves two steps: the measurement update step and the prediction step. The measurement update step integrates the human position into the current belief and the prediction step alters the belief according to the motion model.

The data association is based on the calculated Euclidean distance between the new human cluster and the estimated tracked human. If the distance is less than a threshold value, the new human is associated with the tracked human and inherits its attributes such as ID, appearance time, and velocity. Moreover, suppose that the tracked human disappears due to being occluded or out of range, and the threshold value is adjusted in proportion to the belief uncertainty. For example, the algorithm will search for a new human cluster to connect in a larger region if the human has disappeared for a long time, e.g., five processing steps. In comparison, the search region will be much smaller if the human has just begun to disappear.

The database stores the attributes of each human, including {ID, x, y, vx, vy, appear_time}. Every human has a unique ID and each ID can be assigned to a maximum of two clusters (i.e., two legs). The appear_time attribute keeps track of how long the human has been tracked. The database is updated at each iteration while the location of the target human can be pulled from the database and set as temporal goal in the DWA* navigation system. Each human track is deleted from the database if it disappears more than a fixed period of time. When a human disappears, it could be due to being temporary out of range or occluded. In that case, EKF continues to estimate the locations until it returns to range. However, if a human disappears longer than a pre-set period of time, the track is considered lost and deleted from the database.

The pseudo code of the human detecting and tracking algorithm is depicted in Algorithm 1. If a moving cluster fulfills all three conditions, it is assigned an ID and tracking is initiated using the Kalman filter.

Human Following Algorithm

The goal is to follow a moving target despite the existence of obstacles. A reasonable solution is to see it as a path planning problem and use the obstacle avoidance algorithm to compute possible actions. Compared with the traditional navigation problem, there are two additional challenges in solving a moving target following problem.

1. The robot should be able to act with sufficient agility to react to any drastic change in target motion.
2. Since the target motion mode is changeable and difficult to predict, gathering information is as important as approaching the target. In the planning, the robot should consider its sensing ability in finding a trajectory with maximum information gain.

In this paper, the DWA* navigation system in [16] is adopted as the obstacle avoidance algorithm.

Algorithm 1. Human detecting and tracking algorithm.

Goal: Detect and track humans using 3 conditions
Requires: Moving points in the environment
1: Group moving points into clusters if distance between points < threshold
2: For $i \in \{1, \dots, n\}$, n being the number of clusters segmented
3: If i -th cluster size < 0.5 cm
4: If i -th cluster motion velocity > λ
5: If $ range(t_{start} - 1) - range(t_{start}) > \gamma$ and $ range(t_{end} + 1) - range(t_{end}) > \gamma$
6: Set human_check = true
7: Perform data association
8: Set cluster ID
9: Run Kalman filter updates
10: Update human database
11: End if
12: End if
13: End if

DWA* Navigation System

Dynamic window approach or DWA is a collision avoidance approach strategy developed by Fox et al. [8]. The DWA system is designed to deal with constraints imposed by limited velocities and accelerations when the robot navigates in the environment. DWA* navigation is an extension of DWA proposed in Chou et al. [16]. In the extension system, the A* heuristic [22] is used to search for the optimal trajectory. This approach greatly increases computation speed and enhances overall efficiency.

DWA* is a trajectory-rollout algorithm. First, the environment information is realized as an interval configuration for faster processing. Each interval value represents the maximum distance that can be traveled by the robot on a certain circular trajectory. Second, the intervals are clustered as navigable areas. Third, for each area, a candidate velocity is determined according to an objective function. For each candidate velocity, a new robot position is computed as a new node and is saved in a trajectory tree. Then a node with the minimum estimated cost value will be extracted as the base node for generating new nodes. The procedure is repeated until the goal location is expanded or the tree depth reaches a certain value. After the tree expansion stops, the deepest node is determined as a temporal goal, and the present candidate velocity which can lead the robot to the temporal goal is selected.

Following Algorithm

The primary objective when following the human target is to maintain the target in range while avoiding any obstacles that may block the trajectory. Here, three distinct approaches are applied to achieve optimal human



following performance.

When implementing human following algorithm, an intuitive approach is to set the present location of the moving target as the goal of the navigation algorithm. Considering the tracked target at an angle with respect to the robot, the DWA* algorithm will generate an angular and translational velocity which produces an arc-like trajectory. If the goal is within a close proximity of the robot, the angular velocity will be small and the arc-like route often results in an indirect or detour path for the robot to reach the goal. Consequently, the robot can easily lose the tracked target. The problem lies with the DWA* navigation because the system considers the goal to be a static, non-moving goal in the environment. However, in reality, the goal is a human, moving at an unpredictable velocity and with changing direction. Therefore, the selected DWA* trajectory is no longer optimal because it does not consider where the target goal would be in the next time step.

On the contrary, if the goal is set further away from the robot at the same angle, any movement of the goal will result in a much larger displacement change. Therefore, the angular velocity would increase in response to the substantial change to the goal location. Consequently, the trajectory becomes more directly in line toward the goal.

In the so-called pseudo-goal method, the space in front of the robot is divided into several trajectories, for example, at $-55^\circ, -35^\circ, -20^\circ, 0^\circ, 20^\circ, 35^\circ, 55^\circ$. A pseudo goal is set at three times the original distance between the goal and robot along each trajectory. After acquiring the human location, the pseudo goal with a trajectory closest to the human location is then selected. Setting the goal further away from the robot remedies the issue of the small angular velocity and provides a much more direct path to reach the goal.

Another problem is the limited sensing ability. In this paper, the algorithm is implemented on a mobile robot with a laser range finder with a 180-degree field-of-view, thus providing an information gathering mode. When the moving target enters the 'dangerous zone', i.e., $\pm 55^\circ$, in the robot coordinate, the robot will stop and turn rapidly towards the target. Adding this mode greatly lowers the risk of losing the target.

Although the pseudo-goal method is very simple to implement, it can easily lose track of the target because it does not take the limited observability into consideration. Another drawback is that the pseudo-goal method cannot react quickly to the change of target velocity. A better method is to plan according to a reference trajectory but not a fixed location. In this way, the target velocity can be considered and the robot observability can be considered by adding terms to the cost function. For each possible robot trajectory, a total cost is defined as follows:

$$Cost = \sum_{t=t_0}^{t_0+T} \alpha_1 \cdot dist(t) + \alpha_2 \cdot vis(t) \quad (2)$$

which is a linear combination of two terms: $dist(t)$ depicts the distance between robot and target, and $vis(t)$ reflects whether the robot can see the target at time t . Also, α_1 and α_2 are two constants and T is the evaluating time interval. They are all normalized to [0 1]. When computing $dist(t)$, rather than computing the straight-line distance, another method is used to provide a more reasonable distance estimate based on the fact that, without a straight collision-free path between the robot and the target, the robot must pass a "gap" to reach the target. A gap means a discontinuity in the sensor data and suggests a potentially passable gate. The distance function is modified as follows:

$$\begin{aligned} & \text{If (Straight paths exists between } X_R \text{ \& } X_T) \\ & \quad dist(X_R, X_T) = \|X_R - X_T\|; \\ & \text{else} \\ & \quad dist(X_R, X_T) = \min_g (\|X_T - X_g\| + \|X_g - X_R\|); \end{aligned} \quad (3)$$

where X_R , X_T , and X_g respectively denote the pose information of the robot, the target, and the gap. When performing the heuristic search, the distance metric forces the robot to first search branches which lead to passable gates. As the result, given the same computing time limitation, the robot is more likely to move toward passable gates in early stage and therefore navigates smoothly. Previous studies have shown that, when the distance metric was applied to navigate in a static environment, the performance is better than that of using straight-line distance metric. Computing $dist(t)$ and $vis(t)$ requires repeatedly checking whether a straight path exists between the robot and target. An intuitive solution is the ray-casting algorithm, but this is unsuitable in this task because of its computational complexity.

Algorithm 2 shows the pseudo code to compute distance and visibility costs. The algorithm first acquires the estimated target location or path, depending on whether the pseudo-goal or trajectory-optimization approach is used. It then checks the stop condition and sees if the target human is within one meter. In the pseudo-goal method, the algorithm also checks to see if a rotation is required to maintain the target human within the comfort zone which allows maximum information gathering. Finally, an optimal trajectory is selected using a heuristic search approach and the proper velocities are given to the robot. Algorithm 3 describes the pseudo code of the trajectory-optimization following algorithm.



Algorithm 2. Distance and visibility cost computation algorithm.

<p>Goal: Compute distance and visibility costs</p> <p>Requires: Robot pose, target pose, gap pose</p>
<p>1: Acquire robot pose: XR</p> <p>2: Acquire target pose: XT</p> <p>3: Acquire gap pose: Xg ∈ G</p> <p>4: If straight line exist between XR & XT</p> <p>5: $dist(XR, XT) = XR - XT$</p> <p>6: $v(XR, XT) = 1$</p> <p>7: Else</p> <p>8: $dist(XR, XT) = \min(XR - XT + Xg - XR)$</p> <p>9: $v(XR, XT) = 0$</p> <p>10: End if</p> <p>11: $vis(X_{R,t}, X_{T,t}) = 1 - v(X_{R,t}, X_{T,t}) \cdot \left 1 - \frac{\Delta\theta(X_{R,t}, X_{T,t})}{\pi/2} \right$</p> <p>12: $Cost = \sum_{t=t_0}^{t_{max}} \alpha_1 \cdot dist(X_{R,t}, X_{T,t}) + \alpha_2 \cdot vis(X_{R,t}, X_{T,t})$</p>

Algorithm 3: Human following: trajectory optimization algorithm

<p>Goal: Follow the human target</p> <p>Requires: Target pose, robot pose, target trajectory</p>
<p>1: Acquire target pose and robot pose: XT</p> <p>2: Acquire target trajectory: ET</p> <p>3: While (1)</p> <p>4: $DT = X_T - X_r$</p> <p>5: If $DT > 1$</p> <p>6: Run Distance and Visibility Cost Algorithm: DWA* (ET)</p> <p>7: End if</p> <p>8: End While</p>

Experimental Results and Analysis

In this section, the algorithms proposed in previous sections are extensively tested in designated scenarios and the real-life contexts such as a university library. The proposed algorithms are implemented on a commercial robotic platform, the Pioneer 3 robot, which is a circular differential robot and with a maximum velocity of 0.5m/s. A SICK LMS100 laser range finder with an operating range of 20 m is used.

Robot motion in free space

To compare the pseudo-goal and trajectory-optimization approaches, three different cases of human trajectories are used to test the robustness of robot following: straight line, circular motion, and S shape. The target human trajectories and performed robot trajectories of one tested example are illustrated in Fig. 2. Detailed analyses of these tested experiments are discussed below.

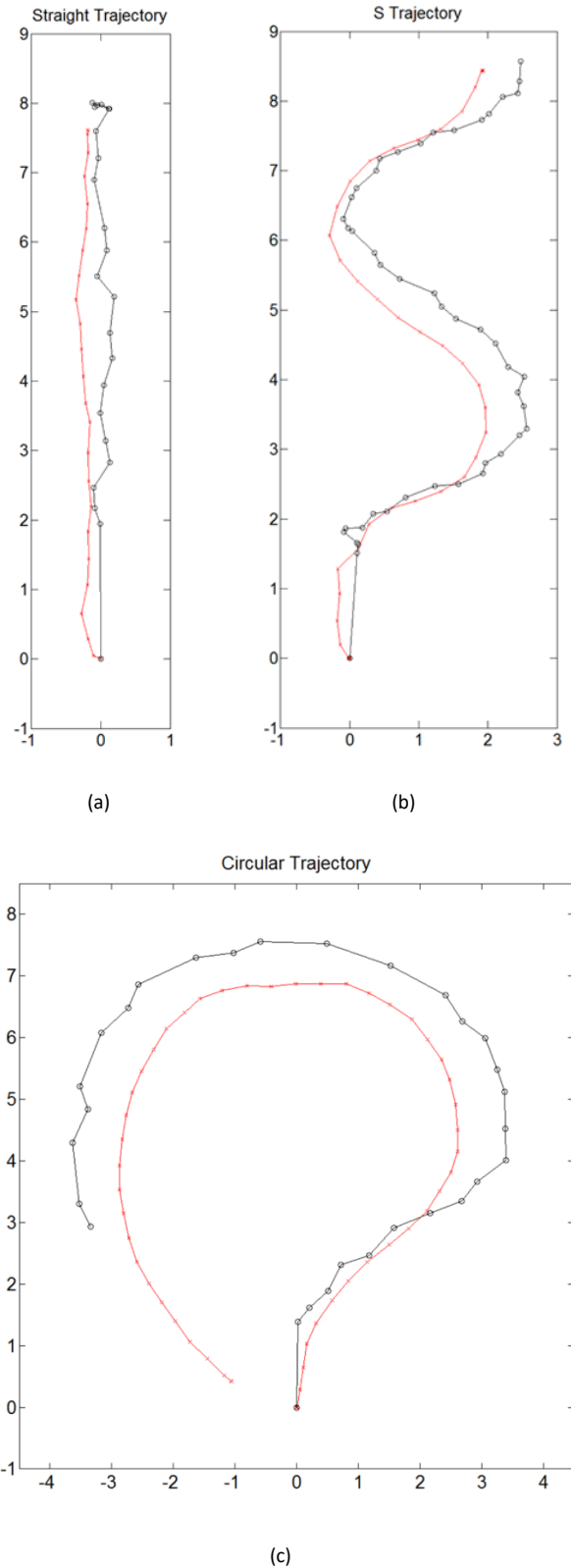


Figure 2: One tested example: (a) straight line, (b) circular motion, and (c) S shape. Note that the target human trajectories and performed robot trajectories are shown in black and red, respectively.

To characterize these maneuvers, three metrics are used to evaluate the experimental performance:

1. Angle between human position and robot pose.



2. Degree of robot fluctuation.
3. Distance between robot and the target.

First, the position and angle of the target human with respect to the robot is computed as follows:

$$\theta = \tan^{-1}\left(\frac{\text{target}_x}{\text{target}_y}\right) * 180 / \pi \tag{4}$$

Ideally, the robot should be able to maintain the target in front (0 degree) at all times to ensure robust following and to maximize target visibility. Directional fluctuation is another metric used to evaluate human following performance. The less the robot’s direction of movement fluctuates, the greater the robot’s stability. The fluctuation degree is evaluated by computing the STD of θ in each trial:

$$STD = \sqrt{\sum_{i=1}^N (\theta_i - \bar{\theta})^2} \tag{5}$$

where N = # of scans.

$$\text{distance} = \sqrt{(\text{target}_x)^2 - (\text{target}_y)^2} \tag{6}$$

In straight-line following, the robot follows a human target in a straight line using local goal approach, pseudo goal approach, and trajectory optimization approach. In each case, three trials are taken in the experiment and the average discrepancy is computed as shown in Table 2. Upon acquiring the average discrepancies, an average value is taken of the three trials as the performance index. In addition, to observe the variation in human angles (i.e., the degree of fluctuation in direction when following the target), the standard deviation is also computed for each trial. The lower the STD value, the less fluctuation, indicating greater robot stability during the experiment. It can be observed that the performance index is further minimized when the trajectory optimization approach is used. In other words, for straight line following, the trajectory optimization approach outperforms the local goal and pseudo goal approaches.

In the case of circular motion following, the robot follows a target human moving in a circular pattern using the pseudo goal approach and the trajectory optimization approach. Note that, when the local goal approach is applied, the robot is incapable of following the target (i.e., loses the target) due to all the detours and delays discussed in the previous section. Table 3 shows the human angle θ and the performance index using the pseudo goal approach. The robot was able to successfully follow the target human. However, the performance indexes are over 30 degrees which indicates the robot had difficulty maintaining the target

in front and often had to perform drastic course corrections to continue the pursuit. However, overall performance is improved by using the trajectory optimization approach.

Table 2. Performance index of straight line following.

Performance Index Metrics	local goal	pseudo goal	trajectory optimization
Average Discrepancy (Degrees)	12.95	10.33	2.84
Standard Deviation (Degrees)	10.70	7.75	3.45
Relative Distance (m)	2.85	2.64	2.37

Table 3. Performance index of circular motion following.

Performance Index Metrics	pseudo goal	trajectory optimization
Average Discrepancy (Degrees)	34.42	28.95
Standard Deviation (Degrees)	35.33	29.41
Relative Distance (m)	3.40	2.54

In S-shape motion following, the target human is instructed to move in an S-shape pattern for the robot to follow. The local goal approach also fails to follow the target human in this case. The S-shape target motion is a more challenging trajectory for the robot to follow since the target changes its velocity abruptly at two separate times. The results are summarized in Table 4, showing a large performance index, similar to the results for circular target motion. Therefore, once again, the robot struggles to maintain the target in range. The computed results indicate that the performance indexes are drastically reduced compared to results from that obtained using the pseudo goal approach. In other words, the robot is much more likely to be facing the target and is also more stable during the experiment. This result is expected since the trajectory optimization approach takes target velocity into consideration when selecting the optimal path and the pseudo goal approach does not. Consequently, it is easier for the robot to adjust its motion by selecting the trajectory that most resembles the target trajectory.

In summary, these experiments show that, when the robot follows a target human in free space, the local goal approach fails for targets moving in a circular or S-shaped pattern. The pseudo goal approach also struggles to maintain the target within the comfort zone. Finally, using the trajectory optimization approach, the discrepancy between the human angle and the robot improves, especially in the case of S-shaped motion, where the human abruptly alters the motion velocity. The variation of the robot motion, illustrated by the standard deviation, is also the smallest for each case when applying



the trajectory optimization approach. In other words, the robot is able to follow the target efficiently and robustly with the trajectory optimization approach. In addition, the robot is capable of following the target closely with a smaller distance between robot and the target when the trajectory optimization approach is applied.

Table 4. Performance index of S-shaped motion following.

Performance Index Metrics	pseudo goal	trajectory optimization
Average Discrepancy (Degrees)	32.30	12.28
Standard Deviation (Degrees)	34.41	13.47
Relative Distance (m)	3.28	3.01

Static and Dynamic Obstacle Avoidance

Another objective of the experiment is to demonstrate the robotic ability to avoid static and dynamic obstacles when following the target. Experiments were performed where the robot follows a target human for a longer duration and also in a narrow hallway. The trajectories and snapshots are shown in Fig. 3. In Fig. 3(a), the rectangle in the map shows the starting position of the robot. The blue circle and line indicates the robot's trajectory.

The cross marks on the map depict the position of the target human during the experiment. The green marks show a secondary human blocking the robot; the robot moves around this person to continue pursuing the target human. The robot rotates to move around the blocking human (snapshot 2) at scan 155 which results in the target human appearing at a less acute angle. Similarly, the robot rotates to avoid static obstacles (boxes shown in snapshot 1) at scan 190. The experiment also shows that the robot is able to follow a target human in a narrow corridor as shown in Fig. 3. Furthermore, implementing the trajectory optimization algorithm best enables the robot to avoid any static or dynamic obstacles through the addition of distance and visibility metrics.

Real-Life Scenarios

As mentioned in the Introduction, one of the primary objectives of achieving human following is to allow robots to assist humans in various environments. After performing the motion maneuver tests and numerous set-up experiments, the proposed algorithms were extensively tested in real-life environments, such as a student cafeteria, a crowded computer center, and a university library. Figure 4 depicts the robot following a human target in a library. In Fig. 4(a), the robot starts to

follow the human target. In addition to the target, two other individual humans appear within robot range as shown in Figs. 4(b) and 4(d). In Fig. 4(c), the target makes a turn into a narrower aisle and the robot continues to pursue. Once again, the target makes another turn into an even narrower aisle between shelves in Figs. 4(f) and 4(h). The robot is able to make the same turn without colliding with the obstacles. The global map is illustrated in Fig. 5. In these real-life scenarios, the robot demonstrates its ability to follow a human target in a more complex and dynamic environment. The robot performed exceptionally well and proved its robustness when the proposed following algorithm is implemented.

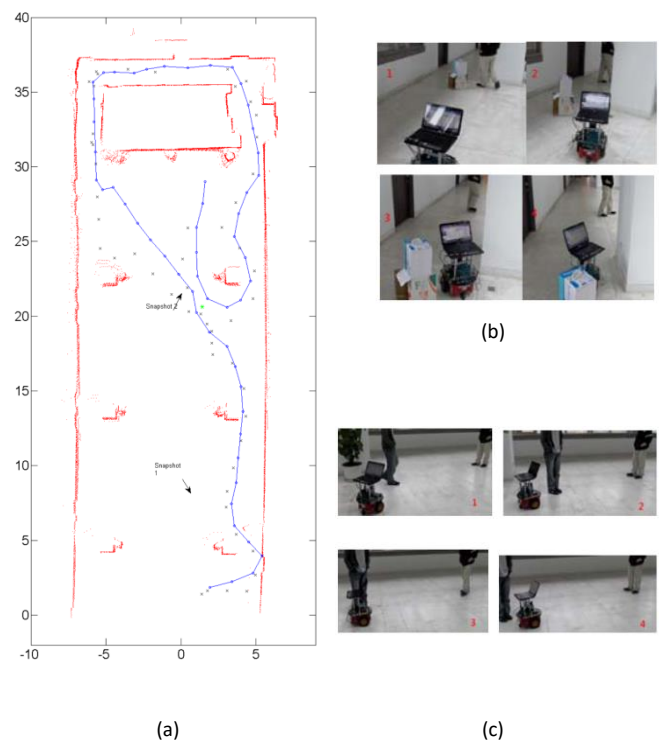


Figure 3. Human Following in Narrow Corridor.(a)Global map showing robot (blue 'o'), target (black 'x'), and human (green '*'). (b) and (c) Snapshots 1 and 2.

Conclusion

This paper presents an active human following system combining DATMO and DWA* to allow a robot to follow a target while avoiding any static or dynamic obstacles along a given path. To improve real-time performance in a dynamic environment, several techniques are used to correct the scan matching results such as outlier filtering, moving objects filtering, and multi-scan matching. These techniques can significantly improve the localization results and still maintain the computational advantage to perform in real time. When detecting humans in the DATMO system, three techniques are proposed to efficiently distinguish and filter out

surrounding noise which can result from the inaccuracy of the laser range finder and more complex environments. Related experimental results demonstrated that applying these techniques significantly reduces noise.



Figure 4. Human following in library.

Furthermore, integration approaches, namely pseudo goal and trajectory optimization, are proposed to combine DWA* navigation with the DATMO system. The pseudo-goal approach does not need to modify the cost function in DWA* and is easy to implement. This allows the robot to follow the target in a smoother and more spontaneous manner by manipulating the temporary goal. However, performance suffers from the lack of consideration of target velocity and trajectory. In other words, the path that robot selects to reach the target is no longer optimal because it does not consider where and how the target will behave in the following instant. Different from the pseudo-goal approach, the trajectory

optimization approach estimates the target trajectory prior to selecting an optimal path. Distance and visibility metrics are implemented in the cost function to choose the optimal path. The distance metric ensures the robot chooses a path that most resembles the target trajectory while the visibility metric maximizes the target visibility at each time instant.

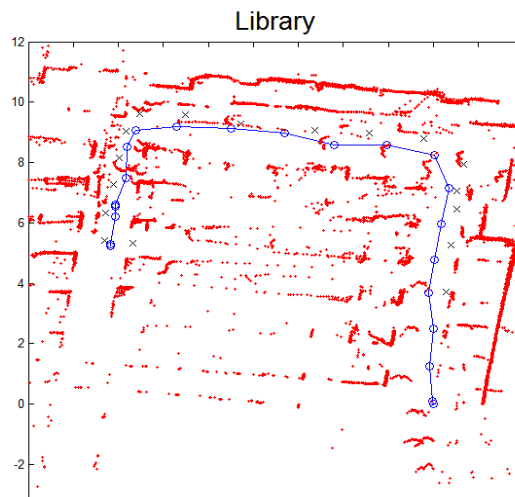


Figure 5. Human following in library. Global map showing robot and target trajectories in blue 'o' and black 'x'.

For future work, there is room for improvement on the DATMO system which can help resolve complex occlusion issues by adopting a more advanced tracking algorithm. As for path planning, a more sophisticated method such as POMDP can also be implemented. Furthermore, a camera can also be equipped on the robot to provide additional features such as color and shape recognition to enhance the robustness of target following and to improve SLAM accuracy. In terms of practical applications, even though the robot has been demonstrated to successfully operate in real-life environments such as a library and cafeteria, additional factors must be considered for full-time service-related tasks in these environments. For example, the batteries for the laser range finder and robot must be recharged regularly. Eventually, the goal is to deploy the robot using the proposed system in a library or office for a longer time to assist humans in various tasks.

Acknowledgment

This work was supported in part by the Ministry of Education, Taiwan, and the Ministry of Science and Technology, Taiwan, under Grants NSC 101-2221-E-002-145- MY3, and NSC 102-2221-E-002-247-MY3.

References

- [1] D. Wolf and G. Sukhatme, "Online simultaneous localization and mapping in dynamic environments," In proceeding of *IEEE International Conference on Robotics and Automation (ICRA)*, New Orleans, LA, USA, April 26-May 1, 2004, vol. 2, pp. 1301-1307.
doi: [10.1109/ROBOT.2004.1308004](https://doi.org/10.1109/ROBOT.2004.1308004)
- [2] L. Jae Hoon, T. Tsubouchi, K. Yamamoto, and S. Egawa, "People tracking using a robot in motion with laser range finder," in proceeding of *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China, Oct. 9-15, 2006, pp. 2936-2942.
doi: [10.1109/IROS.2006.282147](https://doi.org/10.1109/IROS.2006.282147)
- [3] S.-W. Yang and C.-C. Wang, "Simultaneous egomotion estimation, segmentation, and moving object detection," *Journal of Field Robotics*, vol. 28, no. 4, pp. 565-588, 2011.
doi: [10.1002/rob.20392](https://doi.org/10.1002/rob.20392)
- [4] D. Schulz, W. Burgard, D. Fox, and A. Cremers, "Tracking multiple moving targets with a mobile robot using particle filters and statistical data dissociation," in proceeding of *IEEE International Conference on Robotics and Automation*, Seoul, Korea, 2001.
- [5] M. Mucientes and W. Burgard, "Multiple hypothesis tracking of clusters of people," in proceeding of *IEEE/RSJ International Conference in Intelligent Robots and Systems*, Beijing, China, Oct. 9-15, 2006, pp. 692-697.
doi: [10.1109/IROS.2006.282614](https://doi.org/10.1109/IROS.2006.282614)
- [6] X. Shao, Z. Huijing, K. Nakamura, K. Katabira, R. Shibasaki, and Y. Nakagawa, "Detection and tracking of multiple pedestrians by using laser range scanners," in proceeding of *IEEE/RSJ International Conference in Intelligent Robots and Systems (IROS)*, San Diego, CA, USA, Oct. 29-Nov. 2, 2007, pp. 2174-2179.
doi: [10.1109/IROS.2007.4399152](https://doi.org/10.1109/IROS.2007.4399152)
- [7] J. H. Lee, T. Tsubouchi, K. Yamamoto, and S. Egawa, "People tracking using a robot in motion with laser range finder," in proceeding of *IEEE/RSJ International Conference in Intelligent Robots and Systems*, Beijing, China, Oct. 9-15, 2006, pp. 2936-2942.
doi: [10.1109/IROS.2006.282147](https://doi.org/10.1109/IROS.2006.282147)
- [8] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robotics & Automation Magazine*, vol. 4, no. 1, pp. 23-33, 1997.
doi: [10.1109/100.580977](https://doi.org/10.1109/100.580977)
- [9] I. Ulrich and J. Borenstein, "VFH*: local obstacle avoidance with look-ahead verification," in proceeding of *IEEE International Conference in Robotics and Automation (ICRA)*, San Francisco, CA, USA, April 24-28, 2000, pp. 2505-2511.
doi: [10.1109/ROBOT.2000.846405](https://doi.org/10.1109/ROBOT.2000.846405)
- [10] J. Minguez and L. Montano, "Nearness diagram (nd) navigation: Collision avoidance in troublesome scenarios," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 1, pp. 45-59, 2004.
doi: [10.1109/TRA.2003.820849](https://doi.org/10.1109/TRA.2003.820849)
- [11] M. Seder and I. Petrovic, "Dynamic window based approach to mobile robot motion control in the presence of moving obstacles," in proceeding of *2007 IEEE International Conference on Robotics and Automation*, Roma, Italy, April 10-14, 2007, pp. 1986-1991.
doi: [10.1109/ROBOT.2007.363613](https://doi.org/10.1109/ROBOT.2007.363613)
- [12] J. Pineau, G. Gordon, and S. Thrun, "Point-based value iteration: an anytime algorithm for POMDPs," in proceeding of *International Joint Conference on Artificial Intelligence*, Acapulco, Mexico, August 9-15, 2003, pp. 1025-1030.
- [13] N. Roy, G. Gordon, and S. Thrun, "Finding Approximate POMDP solutions Through Belief Compression," vol. 23, pp. 1-40, 2005.
doi: [10.1613/jair.1496](https://doi.org/10.1613/jair.1496)
- [14] C. Sylvie, W. Ong, S. Png, D. Hsu, and W. Lee, "POMDPs for robotic tasks with mixed observability," *In Robotics: Science and Systems*, vol. 5, Seattle, Washington, USA, 2009.
- [15] S. Prentice and N. Roy, "The belief roadmap: efficient planning in linear POMDPs by factoring the covariance," *Robotics Research, Springer Tracts in Advanced Robotics*, Springer, Berlin, Heidelberg, vol. 66, pp. 293-305, 2010.
doi: [10.1007/978-3-642-14743-2_25](https://doi.org/10.1007/978-3-642-14743-2_25)
- [16] C.-C. Chou, F.-L. Lian, and C.-C. Wang, "Characterizing indoor environment for robot navigation using velocity space approach with region analysis and look-ahead verification," *IEEE Transactions on Instrumentation and Measurement*, vol. 60, no. 2, pp. 442-451, 2011.
doi: [10.1109/TIM.2010.2058531](https://doi.org/10.1109/TIM.2010.2058531)
- [17] C. Chih-Chung and L. Feng-Li, "Velocity space approach with region analysis and look-ahead verification for robot navigation," in proceeding of *Joint 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference*, Shanghai, China, Dec. 15-18, 2009, pp. 5971-5976.
doi: [10.1109/CDC.2009.5400887](https://doi.org/10.1109/CDC.2009.5400887)
- [18] J. Pomares, F. A. Candelas, and F. Torres, "Safe human-robot cooperation based on an adaptive time-independent image path tracker," *International Journal of Innovative Computing, Information and Control*, vol. 6, no. 9, pp. 3819-3842, 2010.
- [19] 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 proceeding of *the IEEE International Conference on Robotics and Automation*, Taipei, Taiwan, Sept. 14-19, 2003, pp. 842-849.

doi: [10.1109/ROBOT.2003.1241698](https://doi.org/10.1109/ROBOT.2003.1241698)

- [20] L. Montesano, J. Minguez, and L. Montano, “Modeling the Static and the dynamic parts of the environment to improve sensor-based navigation,” in proceeding of *IEEE International Conference on Robotics and Automation*, Barcelona, Spain, April 18-22, 2005.

doi: [10.1109/ROBOT.2005.1570822](https://doi.org/10.1109/ROBOT.2005.1570822)

- [21] K. Dietmayer, J. Sparbert, and D. Streller, “Model-based object classification and object tracking in traffic scenes from range images,” in proceeding of *IEEE Intelligent Vehicles Symposium*, Tokyo, Japan, 2011.
- [22] A. Stentz, “Optimal and efficient path planning for partially-known environments,” in proceeding of *IEEE International Conference on Robotics and Automation*, San Diego, CA, May 8-13, 1994, pp. 3310-3317.
- doi: [10.1109/ROBOT.1994.351061](https://doi.org/10.1109/ROBOT.1994.351061)

