Submitted on May 04, 2018 as Project Proposal for the Autonomous Intelligent Robotics
Project Proposal CIS 693, EEC 693, EEC793 Autonomous Intelligent Robotics

Robot Companion: With You Always

Mahmoud Hamandi and Labhesh Popli

1 Introduction

With the recent developments in the field of robotics and artificial intelligence, robots are increasingly being assigned tasks where they have to interact with their human host. One interesting task for the robot is to accompany a human to their destination, which can be a lecture hall in a university building, the gate at an airport or to a conference room. In this scope we present an indoor robot service system which will guide the human to their destination in a safe way through crowded areas. We will integrate a Kalman Filter with a motion model based on the social force model (SFM) presented by Helbing and Molar [4] to track humans about the robot. These forces model different aspects of motion behaviors such as the motivation of people to reach a goal, the repulsive effect of walls and other people as well as physical constraints. We present a method to learn the weights of the forces for each of the humans in the environment to be able to track them better. In addition, once we learn the preference of the escortee to follow the robot, we can deduce their intent to follow and give controls to the robot accordingly.

2 Related Work

Luber *et al.* [7] presented a method to track humans based on the Social Force Model (SFM). In [7] they build the motion model describing the human navigation, and track the human with a Kalman Filter with each human's respective motion forces and observations.

On the contrary to the work presented by Ferrer *et al.* [2], [7] assumed that all social forces are equally weighted. While this approach can work for a simulator where the simulated humans are moving according to these equations, it might fail in the real world. Ferrer *et al.* proposed to learn the weight of each force from a collection of human data, and then employ one value online. Although this approach works in [2] where the robot is navigating with the learned social forces, this approach might not work when tracking multiple humans in the real world, as each human might have a different weight for each force.

3 Planned Work

Taking Inspiration from the work done on Beam - a collaborative autonomous mobile service robot [8], our project builds a similar robotic system on Turtlebot (Turtle-Escort), which will escort a visitor from its current position to a given goal location referred to as 'Home'. Due to some unforseen hardware issues suspending any access to the Beam platform, we implemented our system on our lab's (People and Robots Lab) Turtlebot. As Beam is back online in the future the same system will be transfered back to it. For the time being, we have fixed a tripod on the turtlebot to attach a a depth camera at human height. The turtlebot was stabilized by the weight of the laptop placed on top of it, in addition to a small weight placed on one of the tripod's legs as shown in 1. This scheme of robot accompanying the visitor can be useful in scenarios such as a main event in a college or stadium where all humans have to go, while at the same time people visiting need assistance and are not knowledgeable of the space map.

In what follows we will explain the hardware used in this project, as well as detail the algorithms that we will implement.

3.1 Hardware In this project we will require one Turtlebot as the base (available in our lab), as well as a laptop for real-time data processing, an NVidia Jetson TK1, and four IR-depth cameras for human and environment detection. The two processing hardware are required for the real-time processing of the data, as our algorithm requires full observability of the environment. The four depth cameras will help us detect all humans in the scene within the cameras' range.

3.2 Ray Casting to detect Human Distance from Walls To ensure a safe distance of human from walls with respect to sensor readings from robot, we will implement ray casting algorithm and pass it to SFM algorithm to calculate the intent of human.

3.3 Human Detection In this project we require the robot to detect all humans in the scene through the four depth cameras as this information is required for both the intent learning of the human being escorted and ensure the safety and comfort of all humans. Due to the difficulty in accomplishing a detection in all four camera with off the shelf hardware, we decided to detect humans in only one camera, and have the human escort the robot instead of the robot escorting the human. The assumption made here is that the space visible for the robot in one camera is the only part of the environment affecting the human escortee.

Our initial attempt was to use the following ROS packages to do so. The first package [1] extract humans in an image. After extracting the humans in the image, we will identify their location using the depth image. The second package [5] relies on LiDAR scans - provided by the depth cameras in our project- to detect humans legs.

After thorough research Mahmoud realized that [6] provide an open source ROS package that implements many human detectors for any one to test them, as well as a detector that fuse three different human detectors to provide a more accurate final detection, which we rely on in this project.

Figure 1 Figure showing the turtlebot setup. The setup contains the base of the turtlebot, the laptop on top, the tripod holding a depth camera for human detection, a depth camera – not visible– for navigation, and a balancing weight fixed to one end of the tripod.

3.4 Human Tracking and Intent Learning We rely on the social force model presented by Helbing and Molar [4] to track humans and learn their intent. These social forces for a human following a target with a root in the environment can be expressed as:

$$\mathbf{F} = \alpha_0 F_{robot-human} + \alpha_1 F_{human-human} + \alpha_2 F_{obstacle-human} + \alpha_3 F_{target}. \quad (1)$$

Where $\mathbf{F}$ is the resulting force driving the human, $F_{robot-human}$ is the force pushing the human towards or away from the robot, $F_{human-human}$ is the force pushing one human towards or away from other humans, $F_{obstacle-human}$ is the force driving humans away from obstacles, and F_{target} is the force pushing the human to their target. In the case where the target is the actual robot F_{target} would be included in the $F_{robot-human}$. Each of the forces is exponentially related to the distance between the two objects enforcing it, on the exception of the last force that is linearly related to the human speed. αs represent the weight of each force, and it can be considered as the intent of the human to consider this force while navigating. For example, if the human ignores the robot's existence completely, the corresponding α should be zero, if they follow the robot it should be positive, while if they run away from the robot the α should be negative.

Mathematically the forces are represented as follows:

$$F_{k-human} = A_k e^{(d_k - d_{k,human})/B_k} \frac{\mathbf{d_{k,human}}}{d_{k,human}}, \tag{2}$$

where k is one member of the set $\{robot, human, obstacle\}$, A_k, $d_k$and B_k are fixed parameters specific for each member of the set –[2] presents a framework to learn these parameters – $\mathbf{d_{k,human}}$ is the distance vector between the human and the corresponding object and $d_{k,human}$ its norm.
On the other hand:

$$F_{target} = \kappa(\mathbf{v}^0 - \mathbf{v}), \tag{3}$$

where κ is a fixed parameter, $\mathbf{v}^0$ is the desired velocity and $\mathbf{v}$ is the actual human velocity.
While this force is accurate for a moving target, where the human should be matching its velocity with the target's, it does not adapt to the velocity preference of each human when following a fixed target, and does not reflect the difference in direction between the human trajectory and the one leading to the target. Instead we model the force to a fixed target as:

$$F_{target} = \kappa \frac{\mathbf{v}}{v}(1 - cos\theta), \tag{4}$$

where θ is the angle between the direction from the human to the target and trajectory traversed by the human, and v is the norm of the human velocity. This equation places a large emphasis on the difference in direction between the actual trajectory and the one leading to the target, which helps the robot learn the intention of the human to reach a certain target.

For a fixed set of weights, the social force can be calculated for each of the detected humans based on their locations, the robot location, and the location of the obstacles as depicted by the constructed map. Upon calculating the social force we can model the human motion human as presented in [7]:

$$\begin{bmatrix} \mathbf{x_t} \\ \mathbf{v_t} \end{bmatrix} = \begin{bmatrix} \mathbf{x}_{t-1} + \mathbf{v}_{t-1}\Delta\mathrm{t} + \frac{\mathbf{F}}{2}\Delta\mathrm{t}^2 \\ \mathbf{v}_{t-1} + \mathbf{F}\Delta\mathrm{t} \end{bmatrix} \tag{5}$$

where $\mathbf{x_i}$ is the position and $\mathbf{v_i}$ is the velocity of the human at time step i, and Δt is the time difference between the two time frames. Once we have the motion model of each human, we can apply KF localization to find the position distribution of each based on their observation.

Although [2] presented values for the typical αs and [7] calculated them based on average human weight, we cannot ignore that each human navigates with their own based on their personality. In fact, these parameters can change even from one country to the other, or one environment to the other. As such, in this work we suggest to start with a guess about each of the αs and then update them for each human as we receive more observations. Specifically, we can update the parameters from the difference between the expected location of the human based on the estimate of their position with respect to the previous belief and their motion model $P_{x_t|x_{t-1}}$ and the updated position distribution after the observation of their location $P_{x_t|x_t}$. We refer to their difference as $diff(P_{t|t-1}) = h(P_{x_t|x_t} - P_{x_t|x_{t-1}})$, where h(x) extracts the difference in the two resultant forces from the difference in positions.
From $diff(P_{t|t-1})$, we can learn the αs as :

$$\alpha_t = \alpha_{t-1} + diff(P_{t|t-1}).\mathbf{f}.\gamma \tag{6}$$

where **f** is the average vector of individual interaction forces presented in 1 between instances at $t-1$ and t, and γ is the learning rate.

We are mostly interested with two of the interaction weights: the value of α multiplying the *human − robot* force which represents the intention to interact with the robot if the robot is independently navigating, or it would represent the intention to follow or escort the robot if the human and the robot are in an escorting phase.
The second force is the force to target, which can show the interest of the human to reach a certain target. To our knowledge, we are the first in the literature to notice the 'intention' component hidden in the social force model, and we are the first to provide a method to learn these parameters online for each human.

4 Experiments

Unfortunately the SFM algorithm was difficult to test on a real robot although Mahmoud coded a ROS node to calculate the forces and learn the intention weights. However, we were able to test the algorithm in 2-dimensional simulator, similar to the one presented in [3]. The simulator includes a real world environment with humans navigating. The humans' paths are extracted from the ETH walking pedestrian dataset [9].

The experiments conducted where done solemnly to identify if the robot can learn the intention of a human to reach a goal. As such, we picked scenarios where the human is not always going to the target, as in, going to the target part of its trajectory and going away from it in another part.

Labhesh carried out experiments on Turtlebot and used its sensors to check for human legs and detect human distance from the robot. Based on the global position of humans and odometer readings from Turtlebot, updated position was passed to Turtlebot and made him follow the human.

The second experiment was carried out by Labhesh to implement a Ray Casting node to check human distance from the walls, based on the length of rays which are casted on the nearby objects, and calculate the shortest distance from the first collision.

5 Project Outcomes

The expected outcome was to have a fully autonomous system controlled by a human through voice to go to a certain position or follow a human. When following a human, it has to identify the intent of the human to escort, and stop once the human stops escorting. We were able to finalize some parts but not the entire system as shown below.
Mahmoud have coded the human tracking and intention learning in both the 2-dimensional simulator and ROS, and tested it in the simulator.

The code in ROS required a ray tracing software, which Labhesh coded and investigated thoroughly. Labhesh also made ROS nodes for Turtlebot to detect human legs and follow humans to their current position. Although much success was not achieved through ray casting software, thorough research was carried out in the process of understanding. Figure 2 shows example trajectories with the corresponding intention to follow a target. As shown in these images, we can clearly see the value of the intention decreasing slightly in the top part and steeply in the second part, which shows that the algorithm is actually learning that there is a change in the human's intention. Further in these scenarios we have observed that the human tracking gets better over time up until the human direction changes. If we were using fixed weights, we would expect the tracking not to correct itself after that; however, because we are learning the weights online, the tracking fixes itself after observing a few steps as shown in 3.

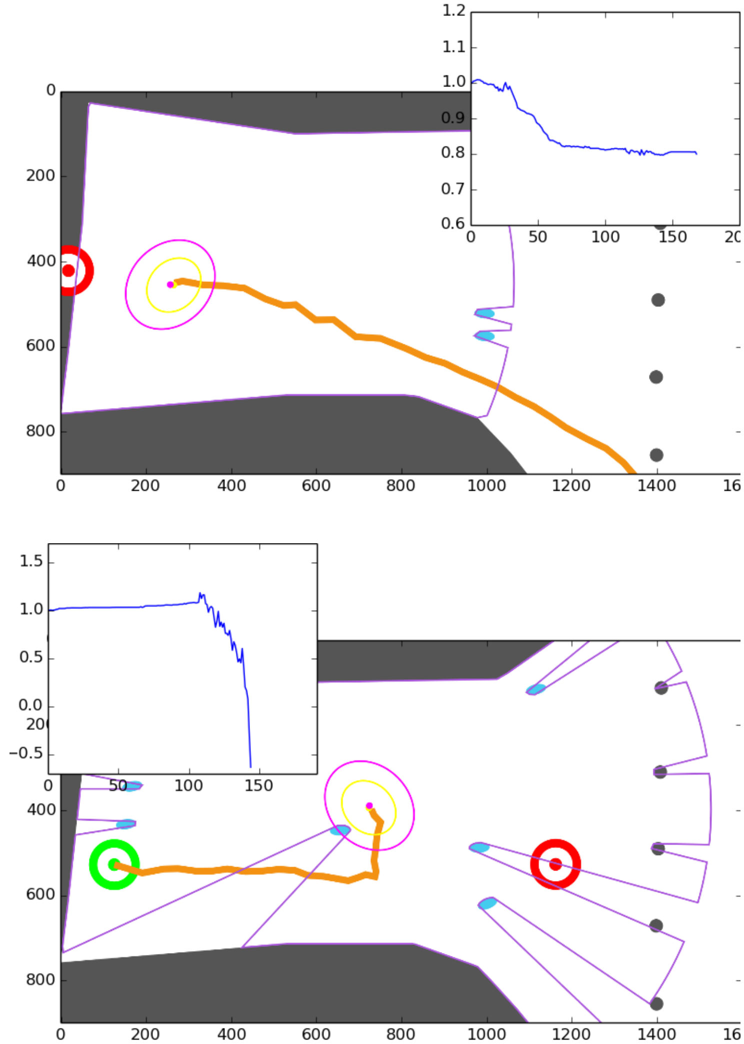

Figure 2 Two trajectories showing a human changing their direction and the corresponding effect on the intention to follow the target. In these images, the yellow line represent the human trajectory, the blue ellipses represent the other humans, the green bulls-eye the starting point, the red bulls-eye the goal target, the purple dot and ellipse the estimate position with its variance, and the yellow dot and ellipse the corrected position and its variance. The graph on top shows the change in goal seeking intention throughout the human path.

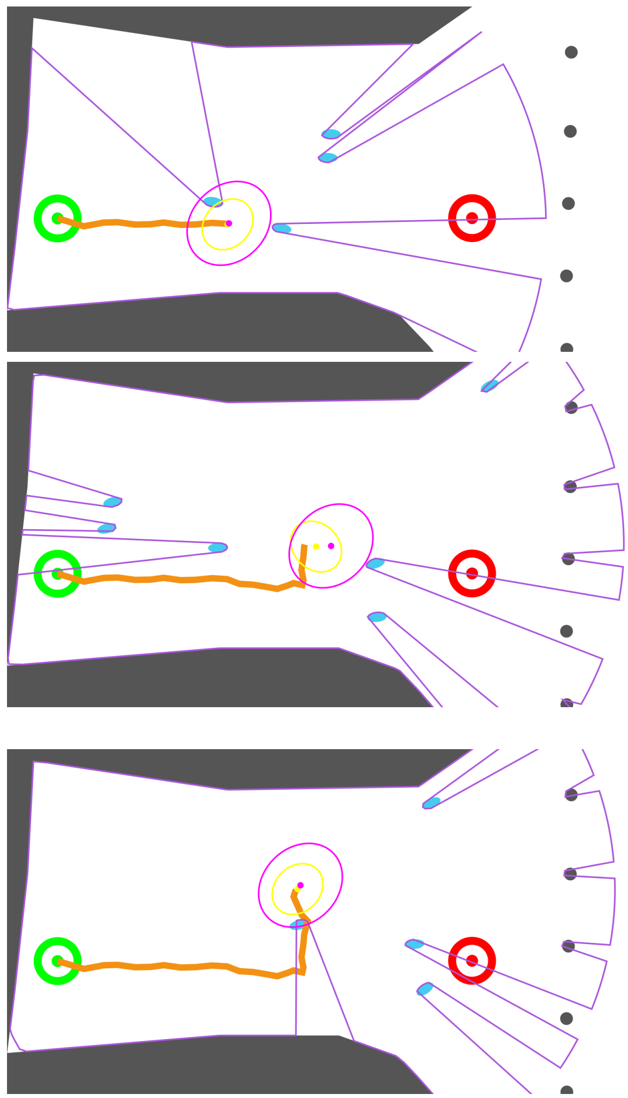

Figure 3 Two trajectories showing a human changing their direction and the corresponding effect on the intention to follow the target. In these images, the yellow line represent the human trajectory, the blue ellipses represent the other humans, the green bulls-eye the starting point, the red bulls-eye the goal target, the purple dot and ellipse the estimate position with its variance, and the yellow dot and ellipse the corrected position and its variance. The graph on top shows the change in goal seeking intention throughout the human path. The three images show in the first the estimated and updated positions to be almost the same, then after the human changes direction the estimated and updated position become far apart, then after we learn the intention of the human we can see the estimated and updated position to be almost the same again.

6 Task Division

Labhesh Popli:Incorporation of the human detection software with leg detection and robot following the human code. Calculating a safe distance of humans in its environment through Ray Casting software implementation and testing. Finalizing the human following part and testing it on the real robot.
Mahmoud Hamandi: Testing of the KF localization, as well as the intent learning in the simulator; furthermore he coded the KF localization and intent learning in ROS, tested the human detector, and made the hardware changes to the turtlebot.
Finally, we were not able to develop the voice command part which was supposed to wrap all parts together.

7 Conclusion

We presented a system for a human to escort a robot to a target location unknown to the robot. The robot detects and tracks the human and follows them to the location, while learning their intent online. Furthermore, we showed the viability of our intention learning algorithm through simulation, which will helps us in future tests to stop the robot automatically when they detect that the human is no longer escorting it.

References

[1] Bormann, R., T. Zwölfer, J. Fischer, J. Hampp, and M. Hägele, "Person recognition for service robotics applications," in *Humanoid Robots (Humanoids), 2013 13th IEEE-RAS International Conference on*, pp. 260–267. IEEE, 2013. 2

[2] Ferrer, G., A. G. Zulueta, F. H. Cotarelo, and A. Sanfeliu, "Robot social-aware navigation framework to accompany people walking side-by-side," *Autonomous robots*, vol. 41 (2017), pp. 775–793. 1, 4, 5

[3] Hamandi, M., M. D'Arcy, and P. Fazli, "Deepmotion: Learning to navigate like humans," *arXiv preprint arXiv:1803.03719*, (2018). 5

[4] Helbing, D., and P. Molnar, "Social force model for pedestrian dynamics," *Physical review E*, vol. 51 (1995), p. 4282. 1, 3

[5] Leigh, A., J. Pineau, N. Olmedo, and H. Zhang, "Person tracking and following with 2d laser scanners," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pp. 726–733. IEEE, 2015. 2

[6] Linder, T., and K. O. Arras, "Multi-model hypothesis tracking of groups of people in rgb-d data," in *the 17th International Conference on Information Fusion (FUSION), 2014*, pp. 1–7. 2

[7] Luber, M., J. A. Stork, G. D. Tipaldi, and K. O. Arras, "People tracking with human motion predictions from social forces," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pp. 464–469. IEEE, 2010. 1, 4, 5

[8] Patel, U., E. Hatay, M. D'Arcy, G. Zand, and P. Fazli, "Beam: A collaborative autonomous mobile service robot," in *AAAI Fall Symposium Series*, 2017. 2

[9] Pellegrini, S., A. Ess, K. Schindler, and L. Van Gool, "You'll never walk alone: Modeling social behavior for multi-target tracking," in *Proceedings of the 12th IEEE International Conference on Computer Vision, ICCV*, pp. 261–268, 2009. 5

Hamandi
Department of Electrical
Engineering and Computer Science
Cleveland State University
Cleveland, Ohio, USA.
Zip code 44115
mahmoudhamandi@gmail.com

Popli
Department of Electrical
Engineering and Computer Science
Cleveland State University
Cleveland, Ohio, USA.
Zip code 44115
labheshpopli@gmail.com