NASA Kennedy Space Center Summer 2019

Nashir A. Janmohamed
Santa Monica College

Major: Computer Science and Engineering
Session: Summer 2019
Date: 16 June 2019
Mentor: Michael DuPuis, NE-L6

Volitional Agent Identification and Path Prediction for Robot Collision Avoidance

I. Introduction

Robot cooperative control, robot-human interaction, and robot navigation in dynamic environments are becoming increasingly prevalent. To achieve maximum utility, robots must learn to de-conflict their trajectories (i.e. avoid collisions) from the trajectories of other independent agents moving in close proximity. This research assumes that if the intent of a volitional agent (VA) can be accurately inferred, so too can greater knowledge be inferred about its future trajectory, thereby improving a control system's ability to plan and execute a collision-avoidance trajectory. A computer vision system will be used to identify VAs in-scene and software will be developed to track and project each agent's trajectory. Information regarding a VA's inferred intent will then be used to augment each projected trajectory.

II. State of the Art

Various methods for generating paths in novel environments currently exist, such as bug algorithms [1, 2], artificial potential field representations [3], and vector field histogram methods [4]. These methods are most effective in static single-agent environments where collision avoidance and path planning only needs to be computed once. However, in environments with multiple volitional agents (VAs), i.e. agents that intentionally choose an action without outside influence, it is necessary to identify other VAs and plan a path accordingly so as to avoid collisions. By using a computer vision system to identify, track and project the paths of VAs, this research has the potential to advance the state of the art in object detection and tracking [5], multi-agent systems (particularly those systems without explicit control over other robots in the system) [6], and robotic pathfinding and trajectory planning [7]. This research has applications to space exploration [8] and in-situ resource utilization (ISRU) [9, 10], but also to terrestrial applications like self-driving cars [11] and autonomous industrial systems [12].

III. Work Done

Before beginning development of the computer vision system, the problem formulation, envisioned end product, and development methodology were discussed at length. It was decided that a top down view, similar to that provided by satellites and other aerial imagery [13], would be used for agent observation. The first iteration of the project uses a 2D top-down view of a simulation with one volitional agent; the second iteration will use a 3D top-down view of a simulation with six volitional agents, the third iteration will use a 3D robot perspective view of the simulation from the second iteration, and the first hardware test of the developed system will be in a First Robotics Competition environment which is a 3D robot perspective view with six volitional agents. Existing simulations with a 2D top down view had four discrete steps – up, down, left, right – or eight discrete steps – the previous four in addition to the four diagonals at 45º. Though all simulations are discrete to a certain degree, limited by the number of pixels in the space representation, the system will be more generalizable to a physical environment if the agent is capable of approximately continuous motion. After looking at existing options for simulations, a custom simulation was developed[1].

The simulation initially had a single volitional agent, obstacles for the agent to avoid, items for it to pick up, and destinations to which the items may be taken. The state of the agent before and after it picks up items is visually indicated by a change in the sprite used. The same is true for the item destinations – after an item has been placed, another item may not be placed in the same item destination. Initially, any destination may be used for a given item, but after a destination has been used, the remaining items must go into distinct destinations. The agent was designed to translate with fixed orientation in order to decrease the complexity of agent detection (discussed later) and allow a greater initial focus on path prediction. The paths of the agents are estimated two different ways, a "conventional method" that projects the path forward by using a polynomial fit, and an "inferred-intent method" that projects the path forward after inferring the agent's intent.

It was decided that the computer vision system would use an open source computer vision library called OpenCV to identify and collect data on the trajectories of agents. Initially, the system was implemented using the Python bindings for OpenCV to expedite the development and prototyping process, but the system will eventually be ported to C++. Python code is interpreted at runtime, whereas C++ is compiled to explicit machine instructions at compile time, and so the decreased overhead of C++ will increase the number of frames per second (FPS) obtained during any given run.

There are many different methods for object detection. One method that is ideal for use in a 2D simulation is template matching, where a template image is compared against overlapped image regions. For each location of **T** (the templated image) over **I** (the region within the image to compare), you store the metric (how well **T** matches with **I**) in the result matrix **R**. Each location (x,y) in **R** contains the match metric, the brightest locations in **R** indicate the highest matches) [14]. Either the best matching image can be reported, or all matches above a certain threshold (i.e. $\mathbf{R}_i >=$ threshold) can be reported. For 3D space representation, in both simulation and physical environments, the Single Shot Multi-box Detector (SSD) [15] approach will be used to detect objects.

Once the volitional agents are detected, their centroids can be computed by using the location reported by the template match (upper left corner) and the known dimensions of the template image. For each frame captured, frame data is saved in a spreadsheet: *time*, *delta_t* between frames (used to determine speed and FPS), *centroid* of the object within the space, *velocity* of the object (a tuple of speed and angle), the *path predicted using conventional means*, and the *path predicted by using inferred intent* of the volitional agent. After the program has finished executing, the data is written to a .csv and all processed frames are written to video.

The "conventional method" uses a linear regression model using the least squares approach to create a polynomial fit to the past trajectory, which is then projected forward in time. The degree of the polynomial depends on the dynamics of the system. Below a minimum, and above a maximum, the trajectory is approximately linear, but between this lower and upper bound, a degree two polynomial is used. In order to fit a polynomial, the points used must fit the definition of a function; if a fit is performed in the original reference frame, there are many cases where the points fail the vertical line test [16]. Accordingly, a linear matrix transformation is performed that uses the current velocity of the agent to reorient the space such that the instantaneous velocity vector is parallel to the x axis in the new frame of reference. This transformation guarantees that at least two points can be used to calculate the polynomial fit (i.e. the last two points that are used to compute the instantaneous velocity). After computing the transformation, the trajectory is extended backwards in time until either the first inflection point, or the first point that fails the vertical line test in the new reference frame.

The "inferred intent method" considers the state of the agent and a goal-oriented hierarchy to infer the most likely destination and then computes an optimal path that it assumes the agent will follow if acting rationally. We assume the agent to be rational, i.e. one that, "for each possible percept sequence (complete history of everything the agent has ever perceived/the state of the agent and the environment), […] should select an action that is expected to maximize its performance measure, given the evidence provided by the percept sequence and whatever built-in knowledge the agent has" [17]. In our case, the percept sequence provides information regarding which goals have been completed and which remain which allows us to infer that the agent will choose the most "rewarding" objective remaining and plan an optimal path. By looking at the current state of the agent and what possible objectives the agent may be trying to complete while observing the rule sets, we can create a goal centered planning hierarchy. We use this hierarchy in addition to the agent path history to project the path of the agent.

At each time step, once all frame data has been computed, the current time, past trajectory, predicted conventional trajectory, and predicted trajectory by inferring intent are plotted on the frame for analysis in post. Predicted trajectories are plotted on frames in real time. Actual trajectory is plotted on frames after collecting all frame data. After completion

[1] Developed by Adam Duke, Embry-Riddle Aeronautical University.

of program execution, the actual location data is compared to the predicted trajectories and used to compute the error at each step between the predicted paths and the actual one. It is assumed that the trajectories computed by inferring intent will be more accurate than those computed conventionally.

IV. Future Work

In the future, this work will be used to predict paths of an intelligent agent trained to complete tasks in simulation. Deep reinforcement learning can be used to train an agent to develop a protocol to complete tasks optimally [18] and large amounts of data can be gathered to validate or show possible improvements in the methods developed as part of this research. As mentioned in the paper, machine learning and deep neural networks have great potential for augmenting the methods presented in this paper, particularly in terms of novel agent identification where pose must be considered. It will also be much more robust in terms of identifying VAs in scene from robot perspective rather than top down. Doing so will require computing distance to other VAs after identifying them, then tracking these VAs through time and space. Research has been done on training agents to infer intent from data [18, 19], and an alternate approach to the work done in this paper would be to find a methodology for path prediction in novel environments using machine learning and neural networks. Lastly, this system can and should inform the GN&C system of a physical robot in order to determine the effectiveness of the system in real-time physical environments.

Acknowledgments

Thank you to NASA, the KSC Education Office, and USRA for providing the opportunity to pursue this research. Thank you to Michael DuPuis for all that you taught me this summer and for being so invested in the success of my project. Thank you to Adam Duke for developing the simulation used for this research. Thank you to Jose Nuñez and Jo Santiago-Bond for welcoming me into your department. Thank you to Kathleen Wilcox, Gwendolyn Gamble, and Rob Cannon for facilitating this once-in-a-lifetime experience.

References

[1] Lumelsky, V.J., & Skewis, T. (1990). Incorporating range sensing in the robot navigation function. *IEEE Trans. Systems, Man, and Cybernetics, 20*, 1058-1069.

[2] Lumelsky, V.J., & Stepanov, A.A. (1990). Path-planning strategies for a point mobile automaton moving amidst unknown obstacles of arbitrary shape.

[3] Khatib, O. (1985). Real-time obstacle avoidance for manipulators and mobile robots. *Proceedings. 1985 IEEE International Conference on Robotics and Automation, 2*, 500-505.

[4] Borenstein, J., & Koren, Y. (1991). The vector field histogram-fast obstacle avoidance for mobile robots. *IEEE Trans. Robotics and Automation, 7*, 278-288.

[5] Yilmaz, A., Javed, O., & Shah, M. (2006). Object tracking: A survey. *ACM Comput. Surv., 38*, 13.

[6] Panait, L., & Luke, S. (2005). Cooperative Multi-Agent Learning: The State of the Art. *Autonomous Agents and Multi-Agent Systems, 11*, 387-434.

[7] Paden, B., Cáp, M., Yong, S.Z., Yershov, D.S., & Frazzoli, E. (2016). A Survey of Motion Planning and Control Techniques for Self-Driving Urban Vehicles. *IEEE Transactions on Intelligent Vehicles, 1*, 33-55.

[8] Pedersen, L., Kortenkamp, D., Wettergreen, D., Nourbakhsh, I.R., & Korsmeyer, D. (2003). A Survey of Space Robotics.

[9] Linne, D.L., Sanders, G.B., Starr, S.F., Eisenman, D.J., Suzuki, N., Molly, S., Anderson, O'Malley, T.F., & Araghi, K.R. (2017). Development for In-Situ Resource Utilization (ISRU).

[10] Mueller, R. P., & Susante, P. J. (2012). A Review of Extra-Terrestrial Mining Robot Concepts. *Earth and Space 2012*.

[11] Litman, T.A. (2015). Autonomous Vehicle Implementation Predictions: Implications for Transport Planning.

[12] Bajoriūnienė, I. (2013). From Automation to Autonomy - a New Trend for Smart Manufacturing.

[13] Gleason, J., Nefian, A.V., Bouyssounouse, X., Fong, T., & Bebis, G. (2011). Vehicle detection from aerial imagery. *2011 IEEE International Conference on Robotics and Automation*, 2065-2070.

[14] Latecki, L. J. (n.d.). *Template Matching*. Lecture presented at CIS 601 Lecture in Temple University, Philadelphia, PA.

[15] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S.E., Fu, C., & Berg, A.C. (2016). SSD: Single Shot MultiBox Detector. *ECCV*.

[16] Stewart, J. (2019). *Calculus: Concepts and Contexts*. Boston, MA, USA: Cengage. p. 17.

[17] Russell, S. J., and Norvig, P., *Artificial intelligence, A Modern Approach*, Upper Saddle River, NJ: Pearson, 2010, pp. 37.

[18] Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., & Riedmiller, M.A. (2013). Playing Atari with Deep Reinforcement Learning. *ArXiv, abs/1312.5602*.

[19] Qi, S., & Zhu, S. (2018). Intent-Aware Multi-Agent Reinforcement Learning. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 7533-7540.

[20] Kitani, K.M., Ziebart, B.D., Bagnell, J.A., & Hebert, M. (2012). Activity Forecasting. *ECCV*.