Gesture-Based Human-Vehicle Leader-Follower Autonomy Systems with a Bipartite Mapping Algorithm

Lawrence Technological University

Research Day, April 22, 2022

Joseph Schulte, MS Computer Science Candidate; Mark Kocherovsky, MS Computer Science Candidate;
Nicholas Paul, MSCS, Adjunct Faculty; Mitchel Pleune, BSCS, CS Robotics Lab Research Staff; CJ Chung, PhD, Professor
Department of Math & Computer Science, College of Arts and Sciences

Lawrence Tech Theory and Practice 1932 Arts and Sciences

INTRODUCTION

In leader-follower autonomy (LFA) systems, autonomous vehicles are given the capability to follow other autonomous vehicles' paths. However, human-vehicle LFA systems had never been demonstrated in the literature until Schulte et. al. in 2022. The Southfield LFA System (SLS), demonstrated on Autonomous Campus Transport 1, a small modified Polaris Gem 2 (See Figure 1), features a modular pipeline made using the Robot Operating System (ROS) to translate human body language into vehicular motion. A camera mounted in the car looking through the windscreen produces a live feed of the activity in front of it. The camera would transmit the frames to an object recognition node powered by Darknet/YOLO to detect the target human and crop and resize the image of the person to the required size for the Pose Estimation module [1-3]. Pose estimation uses a Google-developed pose estimation neural network that translates an image into pose data, which consists of the location and confidence scores of 17 points on the body [3]. The pose data is then fed to a dense neural network of our own design, which produces a predicted command (start: begin following; stop: stop following; none: continue with current behaviour). The prediction is then translated into commands for the vehicle's drive-by-wire system [4].

One issue with the SLS was the loss of target persistence, which occurred when the user was close enough to a third-party that their detections would overlap. The SLS used a closest-match algorithm to figure out which target is correct based on the previous frame. We have improved this using a bipartite mapping method, which ensures that each element in one set is linked to at most one element from another set. In this case, it finds the best match of detections between frames [4, 5].

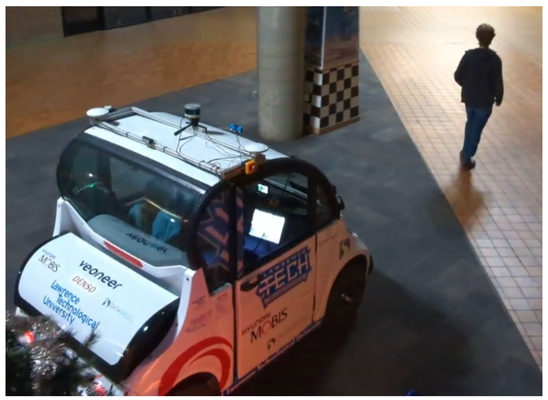

Figure 1: The ACTor1 Platform following author M. Kocherovsky.

REFERENCES

[1] J. Redmon, A. Farhadi. YOLOv3: An Incremental Improvement. arXiv 2018, arXiv:1804.02767
[2] J. Redmon. Darknet: Open Source Neural Networks in C. 2013-2016. Available online: http://pjreddie.com/darknet.
[3] Alphabet Inc. movenet/singlepose/lightning, 2021. Available online: https://tfhub.dev/google/movenet/singlepose/lightning/4.
[4] J. Schulte, M. Kocherovsky, N. Paul, M. Pleune, CJ. Chung Autonomous Human-Vehicle Leader-Follower Control Using Deep-Learning-Driven Gesture Recognition in Vehicles. MDPI. DOI 10.3390/vehicles4010016. (2022).
[5] Gerards, A. M. H. (1995). Chapter 3 Matching. In Network Models (Vol. 7, pp. 135–224). Elsevier. https://doi.org/10.1016/S0927-0507(05)80120-3
[6] F. Chollet. *Deep Learning with Python*, 1st ed. Apress: Berkeley, CA, 2017; Manning
[7] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah,M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, TensorFlow: Large-scale machine learning on heterogeneous systems (2015), software available from tensorflow.org.
[8]. J. O'Kane. *A Gentle Introduction to ROS*. CreateSpace Independent Publishing Platform: 2014; Available online: http://www.cse.sc.edu/~jokane/agitr/

Acknowledgement: Dr. Giuseppe DeRose for his work on ACTor 1 maintenance.

DESIGN

Neural networks are algorithmic structures designed to perform machine learning, i.e. training a model on datasets in order to make predictions on unseen data. We use Tensorflow with Keras to train and implement the model [6, 7]. Communication between components is handled by the Robot Operating System, a platform for distributed control of software and hardware systems using a set of logically connected nodes, which transmit and receive messages as needed [8].

Our training dataset consists of three poses, each with 1700-2000 images. "Start" indicates that the ACTor should start following the user. That user is now designated as the "target". "None" means that the user is giving neither start nor stop commands, and the ACTor will continue with the previous command. If set to follow, the vehicle will track the target (now using bipartite mapping) and attempt to keep within a few meters of the target using the ACTor's LIDAR. If set to stop, it idles and waits for a start command. A "Stop" pose commands the ACTor to end following behaviour if applicable, unmarked the target, and waits for a new "Start" command. Each pose is shown in Figure 2.

Figure 2: Author J. Schulte performing each pose

Our gesture-recognition system uses a modular pipeline of our own design, which is shown in Figure 3. A camera mounted to the front of the car takes a live feed of the user in front of it. Each frame is sent to a node running YOLO/Darknet, a pre-trained object detection system. YOLO returns the shape of the bounding box surrounding each person it detects. These detections are then sent to a node that manages detection matching. Our original system used a simple algorithm to return the closest match between detections in subsequent frames. This is used to ensure that control remains with a single user when following is activated, meaning that a second human cannot inadvertently interrupt the program through errant commands.

However, this led to problems with **target persistence**. If a bystander was close enough in the frame to the target that their detections merged, the targeting system would tend to become confused. This would result in manual program interruption for safety. We thus chose to use a **bipartite mapping algorithm**, which constructs a relationship between two sets where each element in one set is linked to at most one element from another set, demonstrated in Figure 4. The algorithm works by finding a minimal matching of a weighted bipartite graph representing the sets, hence the word "bipartite." When a new frame is captured our algorithm starts by using data from previous frames to predict where each of the detected objects should be in the new frame. It then builds a list of links between the known objects and the detections in the new frame sorted by how close they are to the prediction. The algorithm then goes through each link from closest to farthest and either keeps a link if neither object has been used or discards it if either node has already been linked. The remaining links form a minimal matching, which represents a mapping between existing and new detections, and is used to assign persistent object IDs and perform motion smoothing on all detections, improving the effectiveness and safety of the SLS.

Relevant detections are then cropped, resized, and given to a gesture injection node which contains two key components. The first is a pose estimation model developed at Google which returns the location of 17 body points in the image. Each point also has a confidence score. **This reduces an image with potentially *millions* of data values to a set of 51 points, which is much easier to analyze**. The pose data is then given to a simple neural network of our own design, which returns a command prediction based on the data. This command is then given to a node that interfaces with ACTor 1's drive-by-wire system, which gives movement commands to the vehicle. A diagram of our ROS node architecture is shown in Figure 5, and the diagram of our classifier is shown in Figure 6.

DISCUSSION

We have shown that bipartite mapping can be used to resolve the problem of target persistence in a human-vehicle LFA system. The SLS is now much more robust to interference from bystanders and other potential users. We envision application of the SLS in material transport on factory floors, construction sites, and loading bays. We also see applications in valet parking contexts. In terms of future work, we see the possibility of exchanging our two-dimensional pose estimation model for a three-dimensional model. This would allow us to expand our pose repertoire and mitigate issues with false-positive gesture detections, where the SLS believes that it has detected a pose when none are being shown. We can add this component smoothly into our pipeline due to our modular design paradigm [4].

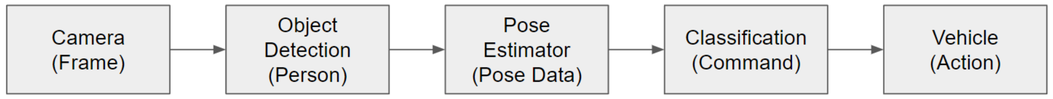

Figure 3: Diagram of the gesture recognition pipeline.

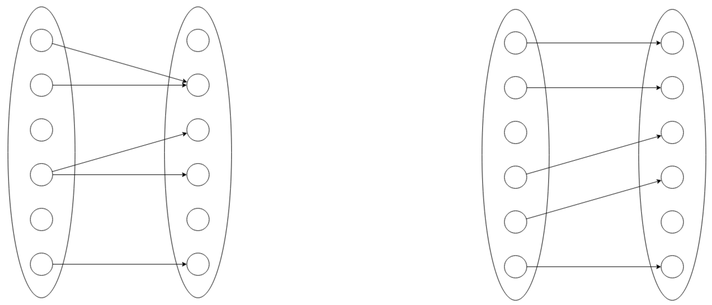

Figure 4: **Left:** a set demonstrating non-bipartite mapping as elements in each set are linked to more than one elements in the other set. **Right:** bipartite mapping is demonstrated because no element is linked to more than one other element.

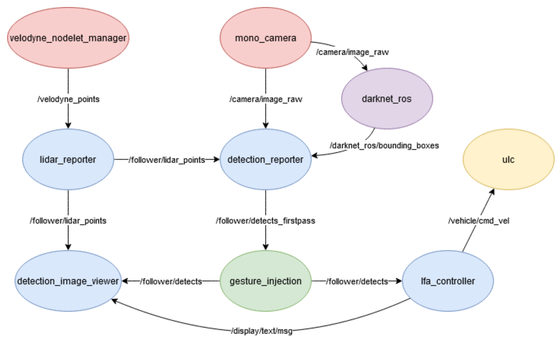

Figure 5: Diagram of ROS node architecture in the SLS

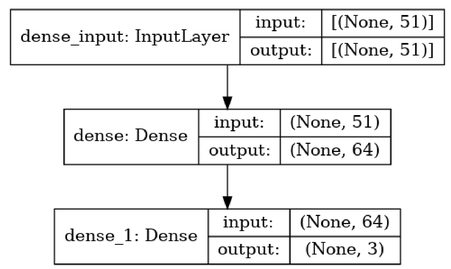

Figure 6: Diagram of our gesture recognition classifier. This does not include object detection or pose estimation modules.

RESULTS

Figure 7 shows the bipartite mapping component in action. First, Authors J. Schulte and M. Kocherovsky prepare to cross each others' paths perpendicular to the frame's angle, i.e passing in front or behind of the other. In the middle image, the crossover occurs. It is clear that both authors are identified separately as persons. Finally, after the crossover, there is no confusion of which person is which. Their identifiers (person 0 or person 1) are the same as they were before. This indicates that the algorithm is working as intended.

Figure 7: **Left:** authors J. Schulte and M. Kocherovsky prepare to cross each other in the camera's view. **Middle:** J. Schulte passes in front of M. Kocherovsky. **Right:** the two separate and their identifiers are unchanged.