Certified Perception for Autonomous Cars

Uriel Guajardo[1], *Annie Bryan*[1], *Nikos Arechiga*[2], *Sergio Campos*[4], *Jeff Chow*[3], *Daniel Jackson*[1], *Soonho Kong*[2], *Geoffrey Litt*[1], *Josh Pollock*[1]

[1]MIT [2]Toyota Research Institute [3]Facebook [4]Universidade Federal de Minas Gerais, Brazil

{annieb22,uriel20,jeffchow,dnj,glitt,jopo}@mit.edu {soonho.kong,nikos.arechiga}@tri.global scampos@dcc.ufmg.br

Abstract

We present a method for establishing confidence in the decisions of an autonomous car which accounts for errors not only in control but also in perception. The key idea is that the controller generates a certificate, which is a kind of proof that its interpretation of the scene is accurate and its proposed action is safe. Checking the certificate is faster and simpler than generating it, which allows for a monitor that comprises a much smaller trusted base than the system as a whole. Simulation experiments suggest that the approach is practical.

1. Introduction

Autonomous cars must be at least as reliable as human drivers, who on average drive 100 million miles between fatal accidents [1]. The environmental complexity of autonomous driving requires large codebases and opaque neural networks that are unlikely ever to be verifiable, and are barely even testable (especially if tests are rerun after any modification).

A classic solution to this dilemma in many cyberphysical systems is to add a runtime monitor whose functionality is limited to preventing accidents, and which, by being smaller and simpler, is amenable to verification. Such a monitor is often called a "safety controller" because its primary function is to enforce control laws. In an autonomous car, however, higher-level planning and control is neither the primary source of complexity nor the cause of common failures. The problem is *perception*: that is, the car does not calculate stopping distances incorrectly but rather fails to notice an obstacle ahead, or misreads the lane markings.

Perception errors can have deadly consequences. For example, in 2018 an Uber automated driving system (ADS) struck and killed a pedestrian walking across the road with their bicycle [2]. This accident was attributable in part to a perception discrepancy. The ADS did not robustly identify the pedestrian, which led to downstream errors in path prediction and ultimately to the crash.

Perception discrepancies underlie many ADS disengagements. In their study of nearly 160,000 ADS disengagements, Boggs et al. [3] found that 21% of them were directly attributable to discrepancies in perception, most of which were object detection errors.

In addition to issues directly attributable to perception, perception feeds information to the planning and control stages [4], and may also be exacerbated by poor weather or road conditions. Such indirect failures account for another 53% of disengagements. All told, perception errors may influence three quarters of all ADS disengagements.

We propose a monitor that can mitigate flaws not only in control but also in perception. The key ideas are:

1. The monitor does not directly access sensors, nor interpret their data, but instead checks a *certificate*, a proof that the proposed action is sound generated by the controller.
2. The certificate uses sensor readings (which are signed and thus unforgeable) and is designed so that the presence, and never the absence, of readings is sufficient to establish confidence in the controller's interpretation.
3. For perception in particular, the certificate includes a small number of properties that can be efficiently checked by the monitor and yet are sufficient to establish a reasonable degree of confidence in the interpretation.

The remainder of our paper outlines the structure of our certificate and its associated checks; describes some experiments we have performed to evaluate the approach in the CARLA simulator [5]; surveys related work; and finally notes some of the key limitations of our approach.

2. A Perception Certificate

Our certificate is designed to mitigate the most likely and most consequential errors in perception: that an object is not registered at all (risking collision with an obstacle the car fails to see); that multiple, distinct objects are incorrectly perceived to be a single object (thus confusing a pedestrian with a car, for example); and that the road surface is not distinguished from potential obstacles.

The controller interprets the scene from a combination of visual and LiDAR data. It constructs a segmentation of the scene into a collection of distinct objects, each labeled with an object type ("car", "pedestrian", etc.) In our implementation, we do not yet make use of the labeling, except to distinguish the road surface from other objects.

The certificate itself includes *only* LiDAR points, because they support a direct physical interpretation. It is structured as (a) a set of point sets, (b) for each point set, a "traversal" comprising list of pairs of points; (c) for each point set, a 3D velocity; and (d) a labeling of the point sets that includes at least whether a point set corresponds to the road surface or another ground plane (such as the sidewalk). Each point corresponds to a single LiDAR reading (a 3D location and 3D velocity relative to the ego car), and can be signed to prevent forgery.

The following tests, corresponding to the errors mentioned above, are applied to the certificate:

1. **Spatial contiguity**. Each pair of points in the traversal for an object is compared to ensure that the points are a maximum distance apart.

2. **Consistent velocity**. Each pair of points in the traversal for an object is compared to ensure that the points have velocities that are compatible within some ϵ_v of the velocity of the point set itself.
3. **Sufficient density**. The 2D space in front of the car is divided into a grid of cells; this test ensures that there is at least one point (from some point set) that falls in each cell.
4. **Ground height**. For any point set labeled as being in the ground plane, the height of every point is checked to ensure that it is indeed at ground level.
5. **Collision distance**. For each point set, we calculate the relative stopping distance using the point set's velocity, the velocity of the ego vehicle, and assumed maximum decelerations, and check that no point in the set is within that stopping distance.

The first two tests ensure that each point set plausibly represents a single physical object, by preventing cases in non-contiguous objects have been merged. The third test ensures that there are no holes in the interpretation of the scene (and requires, incidentally, that areas in the sky or beyond the LiDAR range be represented explicitly as LiDAR points with infinite distance). The final test uses the determined obstacles to prevent collision.

3. Experiments & Evaluation

We are evaluating the approach on scenarios chosen to reflect real conditions. Scenarios tested so far include:

1. The ego car following behind another car;
2. The ego car approaching a stationary car;
3. A bicycle crossing the road ahead of the ego car;
4. A scene with two vehicles ahead of the lead car that are close enough to not be distinguished by the controller.

Scenario (3) was based on the Uber incident in Tempe, AZ). Figure 1 shows a sample frame from the video of this scenario, illustrating the point at which the monitor determines that although the controller has correctly segmented the scene into contiguous objects, the object representing the bicycle is too close given the ego car's speed. In this case, the monitor would issue an emergency braking intervention. The full version of the paper will analyze additional scenarios in more detail.

4. Related Work

Existing approaches to assurance of autonomous systems span a wide spectrum, from formal verification to runtime monitors of controllers. Few, however, address problems with perception. Responsibility-sensitive safety (RSS) [6], for example, has been proposed as a way to provide assurance for the overall autonomy stack, but falls back on statistical notions of correctness to handle sensor errors, and does not attempt to mitigate errors in sensor data analysis.

Various frameworks have been proposed for runtime monitors that provide assurance for controllers, such as the Simplex Architecture [7, 8, 9] and control envelopes [10, 11]. These approaches do not extend to perception systems, since it is not possible to give an input-output specification of their intended behavior.

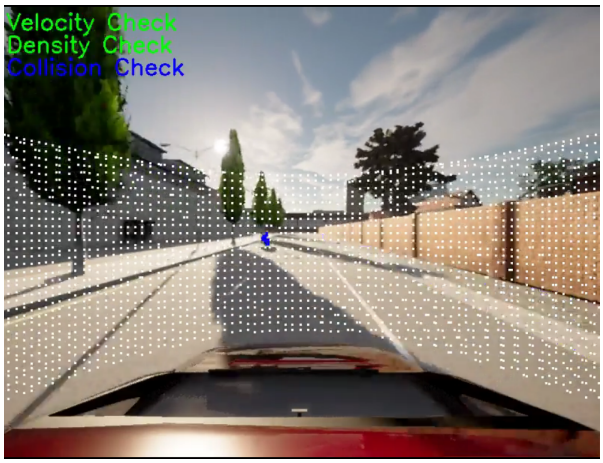

Figure 1: *Collision check failure*

Several approaches have been proposed for formal verification of neural networks [12, 13], but these approaches are either restricted to neural networks used for control or simple properties such as the absence of adversarial examples in a small neighborhood around an input. Our approach, in contrast, can be applied to perception systems beyond local criteria like adversarial examples, and can be extended to those that do not rely on neural networks.

Reasonableness monitors [14, 15] have been proposed as a way to leverage an ontology of a scene in order to detect unexpected perceptions, such as a mailbox crossing a street. Unlike our approach, they do not allow reasoning about fine-grained features of a driving environment, and do not address errors of omission—a major cause of problems. Also, reasonableness monitors will not detect whether objects are incorrectly dropped from the scene.

Credible compilation [16] and Proof-Carrying Code (PCC) [17] have been proposed as general techniques for software components to justify the correctness of their computations. Our approach bears some resemblance to them, but extends them to the case of perception, in which guarantees of correctness that cover all possible cases are not possible.

5. Limitations

Our current checks cannot identify a situation where the controller incorrectly segments a single object into multiple objects, because we only perform checks for object coherence within the bounds of each purported object.

Another limitation is that the density check can spuriously fail for certain object orientations. If an object is at an angle nearly perpendicular to the sensor, the limited resolution of the LiDAR may cause consecutive points in the scan to be measured at far away different distances, causing the check to fail even though the object is actually connected. This is a fundamental limitation of LiDAR information; there is no way to determine based on the sensor data whether the object is fully connected.

Our checks do not currently account for sensor noise and localized failures like omitted points. In the future we could incorporate a more probabilistic approach that takes noise into account.

6. Acknowledgements

This research was funded by the Toyota Research Institute in a collaboration between MIT CSAIL and Toyota. This material is based upon work supported by the National Science Foundation Graduate Research Fellowship Program under Grant No. 1745302.

7. References

[1] S. M. P. Nidhi Kalra. Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability? [Online]. Available: https://www.rand.org/pubs/research_reports/RR1478.html

[2] D. Wakabayashi, "Self-Driving Uber Car Kills Pedestrian in Arizona, Where Robots Roam," *The New York Times*, Mar. 2018.

[3] A. M. Boggs, R. Arvin, and A. J. Khattak, "Exploring the who, what, when, where, and why of automated vehicle disengagements," *Accident Analysis & Prevention*, vol. 136, p. 105406, Mar. 2020. [Online]. Available: https://doi.org/10.1016/j.aap.2019.105406

[4] S. Pendleton, H. Andersen, X. Du, X. Shen, M. Meghjani, Y. Eng, D. Rus, and M. Ang, "Perception, planning, control, and coordination for autonomous vehicles," *Machines*, vol. 5, no. 1, p. 6, Feb. 2017. [Online]. Available: https://doi.org/10.3390/machines5010006

[5] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proceedings of the 1st Annual Conference on Robot Learning*, 2017, pp. 1–16.

[6] S. Shalev-Shwartz, S. Shammah, and A. Shashua, "On a Formal Model of Safe and Scalable Self-driving Cars," *arXiv:1708.06374 [cs, stat]*, Oct. 2018.

[7] T. L. Crenshaw, E. Gunter, C. L. Robinson, L. Sha, and P. R. Kumar, "The simplex reference model: Limiting fault-propagation due to unreliable components in cyber-physical system architectures," in *IEEE International Real-Time Systems Symposium*, 2007.

[8] D. Phan and et. al., "A Component-Based Simplex Architecture for High-Assurance Cyber-Physical Systems," *2017 17th International Conference on Application of Concurrency to System Design (ACSD)*, pp. 49–58, Jun. 2017.

[9] D. T. Phan, R. Grosu, N. Jansen, N. Paoletti, S. A. Smolka, and S. D. Stoller, "Neural Simplex Architecture," *arXiv:1908.00528 [cs, eess]*, Mar. 2020.

[10] N. Arechiga, S. M. Loos, A. Platzer, and B. H. Krogh, "Using theorem provers to guarantee closed-loop system properties," in *2012 American Control Conference (ACC)*. Montreal, QC: IEEE, Jun. 2012, pp. 3573–3580.

[11] N. Arechiga and B. H. Krogh, "Using verified control envelopes for safe controller design," in *American Control Conference*, 2014.

[12] R. Ehlers, "Formal Verification of Piece-Wise Linear Feed-Forward Neural Networks," *arXiv:1705.01320 [cs]*, Aug. 2017.

[13] G. Katz and et. al., "The Marabou Framework for Verification and Analysis of Deep Neural Networks," in *Computer Aided Verification*, I. Dillig and S. Tasiran, Eds. Cham: Springer International Publishing, 2019, vol. 11561, pp. 443–452, series Title: Lecture Notes in Computer Science.

[14] L. H. Gilpin, "Monitoring Opaque Learning Systems," in *ICLR Workshop on Debugging Machine Learning Models*, 2019, p. 8.

[15] L. H. Gilpin and J. C. Macbeth, "Monitoring Scene Understanders with Conceptual Primitive Decomposition and Commonsense Knowledge," *Advances in Cognitive Systems*, p. 20, 2018.

[16] M. Rinard, "Credible compilation," In Proceedings of CC 2001: International Conference on Compiler Construction, Tech. Rep., 1999.

[17] G. C. Necula, "Proof-carrying code," in *Proceedings of the 24th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, ser. POPL '97. New York, NY, USA: Association for Computing Machinery, 1997, p. 106–119. [Online]. Available: https://doi.org/10.1145/263699.263712