INTERNATIONAL ROADMAP FOR DEVICES AND SYSTEMS™

2022 WHITE PAPER

AUTONOMOUS MACHINE COMPUTING

IEEE International Roadmap on Devices and Systems (IRDS) Autonomous Machines White Paper

Shaoshan Liu[1] *Senior Member IEEE*
Jean-Luc Gaudiot[2] *Life Fellow IEEE*

[1]PerceptIn Inc
[2]University of California, Irvine, U.S.A.

Autonomous machines, e.g. autonomous vehicles, are extremely complex systems that integrate many pieces of technologies [1]. For autonomous machines to become an integral part of our daily life, we are still facing many technical challenges. In this white paper, we categorize the technical challenges that are related to devices and systems, introduce the current status and roadblocks, as well as potential research directions.

Area 1: The On-Machine Compute System

As opposed to other computing workloads, autonomous machines have a very deep processing pipeline, or computation graph, with strong dependencies between the different stages and strict deadlines associated with each stage [2]. For instance, Figure 1 presents an overview of the processing pipeline of a level 4 autonomous driving system.

Starting from the left side of the figure, the system consumes raw sensing data from mmWave radars, LiDARs, cameras, and Global Navigation Satellite System (GNSS) receivers and Inertial Measurement Units (IMUs), where each sensor produces raw data at its own frequency: the cameras capture images at 30 FPS and feed the raw data to the *2D Perception module*, the LiDARs capture point clouds at 10 FPS and feed the raw data to the *3D Perception module* as well as the *Localization module*, the GNSS/IMUs generate positional updates at 100 Hz and feed the raw data to the *Localization module*, and the mmWave radars detect obstacles at a rate of 10 FPS. All this raw data is then fed to the *Perception Fusion module*.

Next, the results of the *2D* and *3D Perception Modules* are fed into the *Perception Fusion module* at 30 Hz and 10 Hz respectively to create a comprehensive perception list of all detected objects. The perception list is then sent to the *Tracking module* at 10 Hz to create a tracking list of all detected objects. The tracking list then is fed into the *Prediction module* at 10 Hz to create a prediction list of all objects. After that, both the prediction results and the localization results

Note that this white paper is a shorter but a more technical version of Liu, S. and Gaudiot, J.L., 2022. Rise of the Autonomous Machines. Computer, 55(1), pp.64-73.

are received by the *Planning module* at 10 Hz to generate a navigation plan which then goes into the *Control module* at 10 Hz to generate control commands. These are in turn sent to the autonomous vehicle for execution at 100 Hz.

Hence, every 10 ms, the autonomous vehicle needs to generate a control command to maneuver the autonomous vehicle. If any upstream module, such as the *Perception module*, misses the deadline to generate an output, the *Control module* must still generate a command before the deadline. This could lead to disastrous results as the autonomous vehicle would then be essentially driving blindly without timely participation from the perception unit.

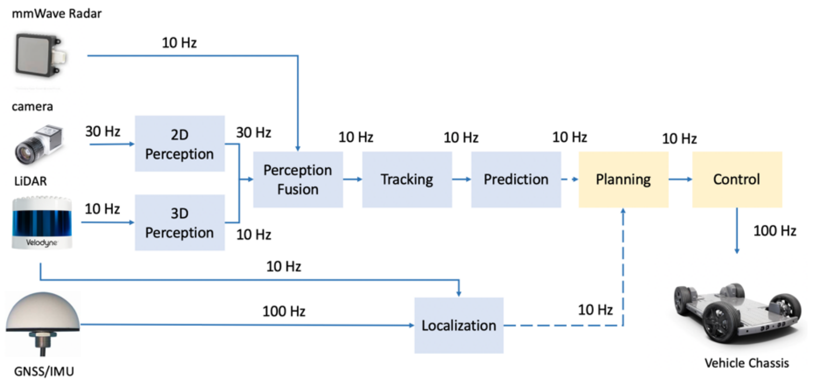

Figure 1: The processing pipeline of an autonomous vehicle

To minimize the end-to-end latency, one commercial approach is to build a proprietary on-machine computing system to map sensing and computing tasks to the best compute substrates to achieve optimal performance and energy consumption [3]. However, this design evolves through trial and error, and the whole design process takes much time with multiple iterations. This is the same approach that many autonomous vehicle companies take: they deploy *ad hoc* solutions to ensure on-time autonomous vehicle product release. These are product-specific and hard to generalize to other autonomous vehicle designs, hence leading to high re-engineering costs for each product.

Challenge: a key technical challenge of designing autonomous vehicle compute system is to develop an appropriate computer architecture, along with a software stack that allows the flexibility of mapping various computation graphs from different types of autonomous vehicles to the same compute substrate, while meeting the real-time performance, cost, and energy constraints. Existing CPUs meet the flexibility requirement but fail to meet the performance and energy constraints, whereas other compute substrates, especially accelerators, typically target to meet the performance and energy constraints of one module, *e.g.,* Perception, without optimizing the end-to-end system.

Area 2: The Perception System

Perception is essential to any autonomous machine applications where sensory data and artificial intelligence techniques are involved. The final objective of perception is to extract spatial and semantic information from the raw sensing data so as to allow the machine to construct a comprehensive understanding of its operating environment. Such understanding includes the types, positions, headings, speeds, and dimensions of all objects in the environment.

There are two categories of perception for autonomous machines, the deep learning-based approach and the geometry-based approach. The *deep learning-based* approach is mainly used to extract semantic information and is heavily used in applications such as object detection, scene understanding, segmentation, tracking and prediction, *etc*. Within an autonomous machine's perception system, multiple deep learning models are running simultaneously, *e.g.,* networks for 2D perception, networks for 3D perception, and networks for tracking and prediction. Nonetheless, deep learning-based perception is often the performance bottleneck in autonomous machines. It thus becomes a painful tradeoff between perception quality and compute and energy resource utilization.

The *geometry-based* approach is mainly used to extract positional and dimensional information of the target objects. A most common instance of geometry-based perception application is the real-time stereo vision used for autonomous machine navigation, obstacle avoidance, and scene reconstruction. Stereo vision allows autonomous machines to obtain 3D structure information of the scene. A stereo vision system typically consists of two cameras to capture images from two points of view. Disparities between the corresponding pixels in two stereo images are detected using stereo matching algorithms. Depth information can then be calculated from the inverse of this disparity. Autonomous machines must fuse the positional and dimensional information and the semantic information of the target objects to form a comprehensive understanding of their environments.

Challenge: a key technical challenge for perception is in the development of a general framework to generate reliable and precise understanding of the operating environment in real time. Reliable and precise perception can be achieved by fusing various perception results, such as 2D semantics, 3D semantics, and 3D geometry, an area still currently under active research. On the other hand, to achieve real-time performance, software approaches such as the compression-compiler co-design method that combines the compression of deep learning models and their compilation to optimize both the size and speed of deep learning models [4, 5]. Hardware approaches such as hardware accelerators for perception modules [6, 7, 12] can be taken. Ultimately, more research is required to determine what combination of software and hardware approaches will be most effective in achieving real-time performance for the perception system.

Area 3: The Localization System

Fundamental to autonomous machines is localization, *i.e.,* ego-motion estimation, which calculates the position and orientation of an agent in each frame of reference. Formally, localization generates the six degrees of freedom (DoFs) pose, including the three DoFs for the translational pose, which specify the <x, y, z> position, and the three DoFs for the rotational pose, which specify the orientation about three perpendicular axes, *i.e.*, yaw, roll, and pitch. Knowing the translational pose fundamentally enables an autonomous machine to plan its path and to navigate, while the rotational pose further lets it stabilize itself.

Localization is highly sensitive to the operating environment, as different environments require different sensors and algorithms. For instance, in outdoor environments which usually provide stable GNSS signals, the compute-light visual inertial odometry (VIO) algorithm or LiDAR odometry coupled with GNSS signals achieves the best accuracy and performance. In contrast, in unknown, unmapped indoor environments, a LiDAR or visual Simultaneous Localization and Mapping (SLAM) algorithm delivers the best accuracy [8, 11]. Nonetheless, different localization algorithms often incur different latencies, and even worse, latency variations. For instance, for visual odometry or visual SLAM, the processing latency often depends on the number of feature points extracted from the current image, which means large latency variations that might impact predictability and safety.

Challenge: a key technical challenge in localization is to develop a standard framework capable of adapting to different operating scenarios by unifying core primitives in various localization algorithms, and to seamlessly switch between different algorithms as the autonomous machine navigates through different environments. In addition, this standard framework should provide a desirable software baseline for acceleration which will minimize processing latency as well as latency variations.

Area 4: The Planning and Control System

The planning and control system dictates how an autonomous machine should maneuver. Traditional planning and control systems must include behavioral decisions, motion planning and feedback control kernels [1]. Specifically, motion planning entails three steps, namely roadmap construction, collision detection, and graph search.

While commonly deployed in commercial autonomous machines, traditional planning and control methods often utilize human-in-the-loop rule-based approaches, where engineers fine tune the planning and control kernels with available test data. This approach is not only slow and costly, but also not robust as all “corner” cases need to be added manually. As more rules must be included, the planning and control system becomes increasingly large and unwieldy and often fails to meet real-time requirements. In addition, rule-based methods suffer from a notorious difficulty to handle the multi-agent problem, or the fact that the actions taken by an agent can affect the behavior of other machines in the same environment, hence failing to handle complex traffic scenarios.

Recently, the Deep Reinforcement Learning (DRL) for planning and control is being actively researched in many places worldwide. Compared to traditional planning and control methods, inference with DRL incurs low computational requirements during operation, especially for high degree of freedom configurations [9]. In addition, DRL methods are capable of handling the multi-agent problem, allowing autonomous machines to handle complex traffic scenarios. However, model training is the bottleneck for DRL-based planning and control systems, as model training requires a vast number of trials to gain enough experience. This is especially the case for complex scenarios where model training can easily reach millions of steps, with each setup of hyper-parameters or reward hypothesis taking hours or even days.

Challenge: while DRL-based planning and control methods are highly promising, a key technical challenge for planning and control remains the development of a cloud infrastructure that provides sufficient compute power and generates high-quality data for DRL model training. First, to ensure we generate enough high-quality data to train the DRL networks, we need to develop simulation engines that are capable of closely simulating various physical scenarios. Second, to greatly improve algorithm development efficiency, especially for complex scenarios, we need to develop a model training infrastructure that can reduce the training time by orders of magnitudes.

Area 5: The Cooperation among Autonomous Machines

While traditional autonomous machines utilize only on-machine intelligence, cooperative autonomous machines depend on the cooperation between autonomous machines in addition to the infrastructure. Take autonomous driving for example, the infrastructure-vehicle cooperative autonomous driving approach relies on the cooperation between intelligent roads and intelligent vehicles. This approach is not only safer but also more economical compared to the traditional on-vehicle-only autonomous driving. Based on the progress towards commercial deployment, a three-stage development roadmap has been proposed as follows [10]:

- Stage 1: infrastructure-augmented autonomous driving (IAAD), in which autonomous vehicles fuse vehicle-side and infrastructure-side perception outputs to improve safety of autonomous driving.
- Stage 2: infrastructure-guided autonomous driving (IGAD), in which autonomous vehicles can offload all the proactive perception tasks to the infrastructure in order to reduce per-vehicle deployment costs.
- Stage 3: infrastructure-planned autonomous driving (IPAD), in which the infrastructure takes care of both perception and planning, thus achieving maximum traffic efficiency and cost efficiency.

To complete these tasks, Figure 2 presents an overview of the cooperative autonomous driving system architecture. It consists of the Systems on Vehicle (SoVs), the Systems on Road (SoRs), the intelligent transportation cloud system (ITCS), and the control center. The SoRs provide local perception results to the SoVs for blind spot elimination and extended perception to improve safety. Meanwhile, the SoRs process incoming sensor data and send the extracted semantic data to the ITCS for further processing. The ITCS fuses all incoming semantic data to generate global perception and planning information before the control center can dispatch real-time global traffic information, navigation plans, and even vehicle control commands to the SoVs to achieve optimal traffic efficiency.

Challenge: a technical challenge for cooperative autonomous machines is integration, as its full realization relies on all the technical areas described in the previous sections. To guide the progress of cooperative autonomous vehicles, the community needs to clearly define the technical specifications and standards of each technical area for each stage of deployment so as to ensure effective technical integration.

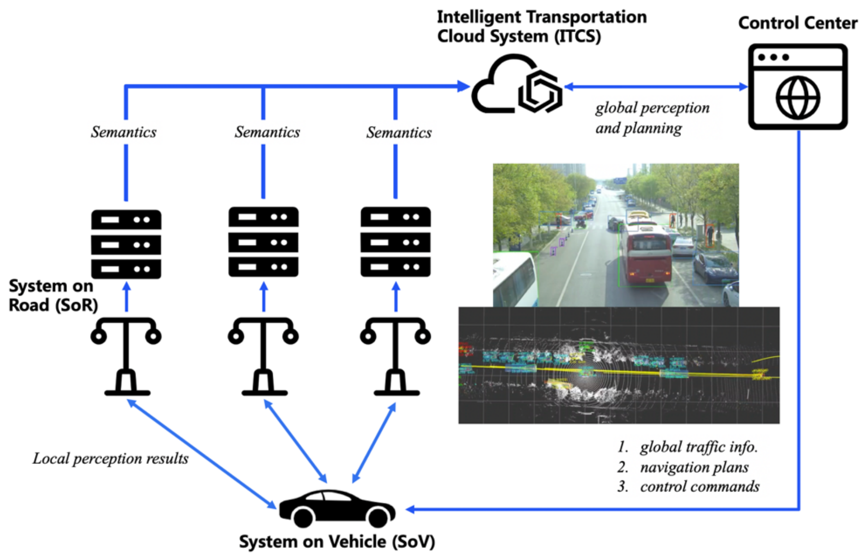

Figure 2: An infrastructure-vehicle cooperative autonomous driving system

Summary

After more than six decades of information technology development, we believe that autonomous machines will completely revolutionize our daily life and our economy in the coming decade. The impact of autonomous machines on our society is likely to be much deeper and broader than any other information technology revolution that we have experienced in the past decades. To facilitate the rise of autonomous machines, in this white paper we have categorized the technical challenges that are highly related to electronic devices and systems.

References

1. Liu, S. and Gaudiot, J.L., 2022. Rise of the Autonomous Machines. Computer, 55.
2. Liu, S., Tang, J., Zhang, Z. and Gaudiot, J.L., 2017. Computer architectures for autonomous driving. Computer, 50(8), pp.18-25.
3. Yu, B., Hu, W., Xu, L., Tang, J., Liu, S. and Zhu, Y., 2020, October. Building the computing system for autonomous micromobility vehicles: Design constraints and architectural optimizations. In 2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)
4. Guan, H., Liu, S., Ma, X., Niu, W., Ren, B., Shen, X., Wang, Y. and Zhao, P., 2021. CoCoPIE: enabling real-time AI on off-the-shelf mobile devices via compression-compilation co-design. Communications of the ACM, 64(6), pp.62-68.
5. Zhao, P., Niu, W., Yuan, G., Cai, Y., Sung, H-H., Liu, S., Liu, S., Shen, X., Ren, B., Wang, Y., Lin, X. Brief Industry Paper: Towards Real-Time 3D Object Detection for Autonomous Vehicles with Pruning Search. In 2021 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS).

Note that this white paper is a shorter but a more technical version of Liu, S. and Gaudiot, J.L., 2022. Rise of the Autonomous Machines. Computer, 55(1), pp.64-73.

6. Gao, T., Wan, Z., Zhang, Y., Yu, B., Zhang, Y., Liu, S. and Raychowdhury, A., 2021. ielas: An elas-based energy-efficient accelerator for real-time stereo matching on fpga platform. In 2021 IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS).
7. Wan, Z., Yu, B., Li, T.Y., Tang, J., Zhu, Y., Wang, Y., Raychowdhury, A. and Liu, S., 2021. A survey of fpga-based robotic computing. IEEE Circuits and Systems Magazine, 21(2), pp.48-74.
8. Gan, Y., Bo, Y., Tian, B., Xu, L., Hu, W., Liu, S., Liu, Q., Zhang, Y., Tang, J. and Zhu, Y., 2021, February. Eudoxus: Characterizing and Accelerating Localization in Autonomous Machines Industry Track Paper. In 2021 IEEE International Symposium on High-Performance Computer Architecture (HPCA)
9. Kiran, B.R., Sobh, I., Talpaert, V., Mannion, P., Al Sallab, A.A., Yogamani, S. and Pérez, P., 2021. Deep reinforcement learning for autonomous driving: A survey. IEEE Transactions on Intelligent Transportation Systems.
10. Liu, S., Yu, B., Tang, J. and Zhu, Q., 2021. Invited: Towards Fully Intelligent Transportation through Infrastructure-Vehicle Cooperative Autonomous Driving: Challenges and Opportunities. In Proceedings of the 58th Annual Design Automation Conference (DAC)
11. Liu, W., Yu, B., Gan, Y., Liu, Q., Tang, J., Liu, S. and Zhu, Y., 2021, October. Archytas: A Framework for Synthesizing and Dynamically Optimizing Accelerators for Robotic Localization. In MICRO-54: 54th Annual IEEE/ACM International Symposium on Microarchitecture (pp. 479-493).
12. Liu, S., Wan, Z., Yu, B. and Wang, Y., 2021. Robotic computing on fpgas. Synthesis Lectures on Computer Architecture, 16(1), pp.1-218.

Author Information:

SHAOSHAN LIU is the founder and CEO of PerceptIn Inc, an intelligent transportation company. Shaoshan Liu's technical research focuses on Autonomous Driving technologies and Robotics. He has published 90 research papers, 40 U.S. patents, and over 150 international patents. Dr. Shaoshan Liu have published three books on robotics technologies "Creating Autonomous Vehicle Systems (Morgan & Claypool)", "Engineering Autonomous Vehicles and Robots: The DragonFly Modular-based Approach (Wiley - IEEE)", and "Robotic Computing on FPGAs (Morgan & Claypool)" Dr. Shaoshan Liu was a founding member of Baidu USA, where he built the Autonomous Driving Systems team, and performed R&D work at LinkedIn, Microsoft, Microsoft Research, INRIA, Intel Research, and Broadcom. He is a senior member of IEEE, a Distinguished Speaker of the IEEE Computer Society, a Distinguished Speaker of the ACM, and a founder of the IEEE Special Technical Community on Autonomous Driving Technologies. Dr. Shaoshan Liu received Master of Public Administration (MPA) from Harvard Kennedy School, Ph.D. in Computer Engineering from University of California, Irvine. Contact him at shaoshan.liu@perceptin.io

JEAN-LUC GAUDIOT is a professor in the Department of Electrical Engineering and Computer Science at the University of California, Irvine, California, 92697, USA. His research interests include multithreaded architectures, fault-tolerant multiprocessors, and the implementation of reconfigurable architectures. Gaudiot received a Ph.D. degree in computer science from the University of California, Los Angeles. Gaudiot served as the 2017 IEEE Computer Society President. He is a Fellow of IEEE and the American Association for the Advancement of Science. Contact him at gaudiot@uci.edu

Note that this white paper is a shorter but a more technical version of Liu, S. and Gaudiot, J.L., 2022. Rise of the Autonomous Machines. Computer, 55(1), pp.64-73.