Perception: The Most Difficult Technical Challenge for Automated Driving

Improving Real-Time Perception with Deep Sensor Fusion

Forrest Iandola, PhD, CEO - DeepScale
Ben Landen, MBA, Director of Business Development - DeepScale

Today's mass-produced driver-assistance systems are typically implemented with a Late Fusion paradigm, which is described as follows: each camera or radar sensor has an onboard processor that runs perception software to generate an object list. Then, the object lists are sent to an Electronic Control Unit (ECU), which performs probabilistic techniques to merge the object lists. This late-fusion approach has a number of limitations in terms of accuracy, portability, and robustness to sensor failure. We propose an earlier stage of fusion, called Deep Sensor Fusion, where sensors transmit raw data over higher bandwidth in-vehicle networking already used in mass production, and this data is processed by a modern ADAS Domain Controller. We show how Deep Neural Networks (DNNs) can ingest raw data from multiple types of sensors and generate improved perception results. Finally, we implement DNNs in real-time using automotive-grade processors to enable all SAE levels of automated driving.

Building Blocks of Automated Driving

The automated driving "flow" can be simplified into a few high-level blocks:

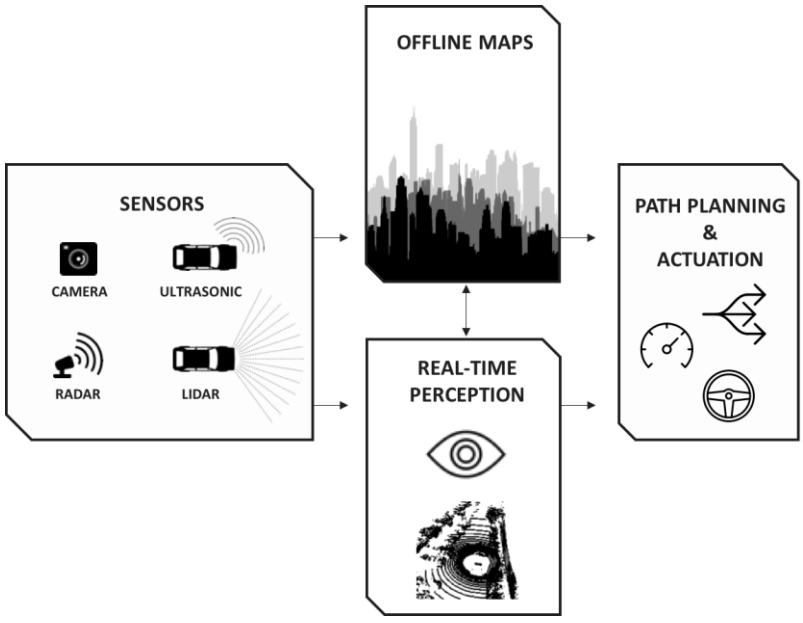

Figure 1: Automated Driving Flow

This diagram captures the fundamental components of an automated driving system, but the simplicity of the diagram does not do justice to the complexity of each block and its various sub-

systems. Furthermore, integrating these blocks into a cohesive system that can be deployed at scale with commercial viability is non-trivial yet instrumental for adoption of automated driving.

System technical requirements increase as vehicles aim to accomplish higher levels of automation as defined by SAE [1]. Subsequently, each of the building blocks must improve its respective performance or risk becoming the weakest link and derating the capability of the entire system. While not trivializing any of the unique technical challenges in the flow, industry experts implementing the full stack of automated driving are consolidating on the viewpoint that perception is the most difficult problem to solve. DeepScale is solving the technical and implementation challenges of real-time perception.

Perception in Real-World Driving

An August 2017 interview with Brian Salesky (CEO of Argo AI, which is chartered by majority stakeholder Ford Motor Company to develop Ford's first fully autonomous vehicle) stated:

> *Salesky, along with many others in the field, say perception is the stickier problem, because it's important for the autonomous vehicle to not only detect relevant objects, but to predict what those objects like a car, pedestrian, or bicyclist are going to do. Once it has the correct and robust information, making a decision is relatively easy.* [2]

The corollary is that automated driving in simulations is a solved problem because the simulator has perfect information about the environment. The perfect environmental information translates to straightforward decisions for path planning and vehicle actuation. These decisions maintain high fidelity when applied to real vehicles by accounting for classical mechanics. The gap exists in the fact that we have not yet solved the problem of extracting perfect environmental information with automated driving systems that build perception from sensors, silicon, and software.

Late Sensor Fusion

Today's automotive perception implementations take a sensor-centric approach. Unique perception software is developed for each sensor, and the outputs from each of these siloed sensors get combined to provide the data that ultimately informs the driving system what actions it should take. Several mass-market vehicles that are on the road today use this type of perception system for Advanced Driver Assist Systems (ADAS) like Automatic Emergency Braking (AEB) and Adaptive Cruise Control (ACC). Such a system might look like this:

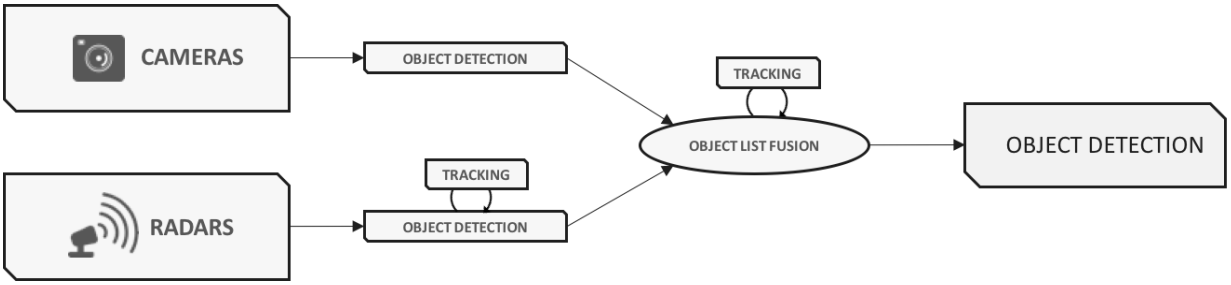

Figure 2: ADAS Late Fusion Perception System

This architecture works reasonably well for simple ADAS functions like AEB and ACC that are still mostly confined to highway driving in mass production deployments. For these functions, the system just needs to figure out if there is an object (for highway driving, the concern is limited mostly to other vehicles) in the vehicle's trajectory that presents a crashing risk. If there is danger of a collision, the brakes are applied.

Problems arise when this Late Fusion architecture is expanded to more advanced driving systems that aim to offer more complicated driving maneuvers and must detect additional object classes, infer accurate distances, trace lanes, read traffic lights—the list of perception requirements goes on. As more sensors are introduced with their own siloed software to enable these new capabilities, the system quickly reaches unwieldy complexity:

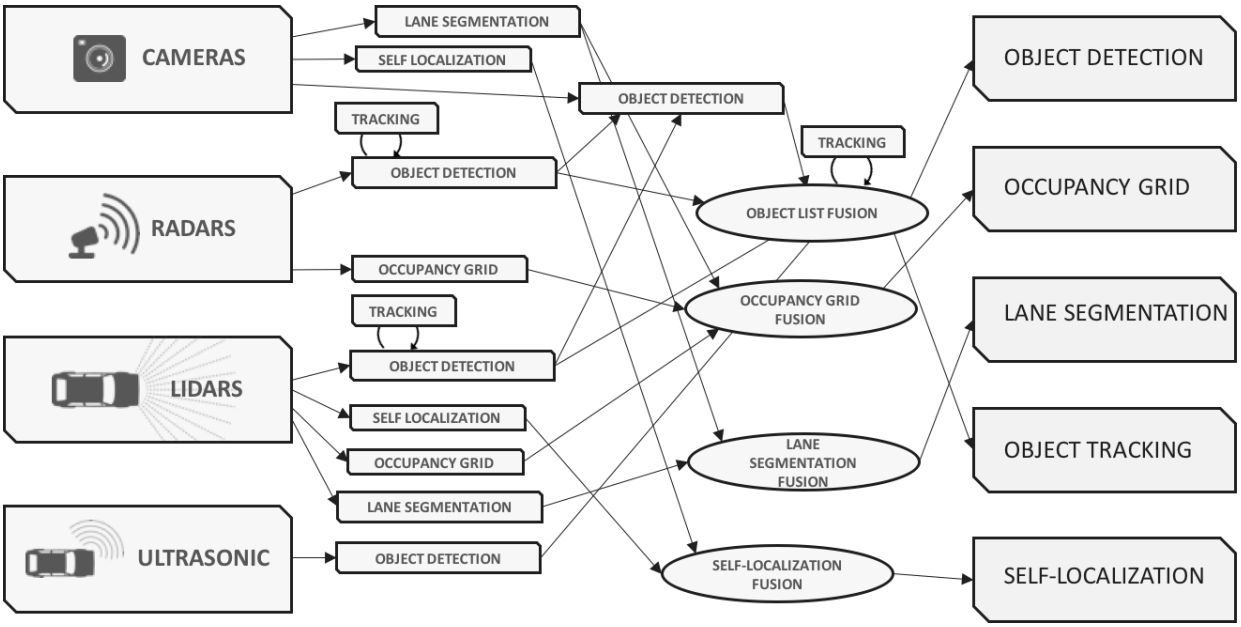

Figure 3: Autonomous Vehicle Late Fusion Perception System

The expanded system requirements exacerbate a few prevalent shortcomings of Late Fusion:

1) Fusing processed outputs from different sensors throws away critical information that can improve accuracy and cover corner cases. If sensors disagree, the system must take a best guess at which sensor is correct.
2) Creates a sensor hierarchy that limits system performance to being only as good as its best sensor. Failure of a single sensor can render the whole system unusable.
3) Creates complicated software implementations. Sensor-specific software must be re-written for each new model of sensor that is released. Each fusion algorithm is written based on specific sensor inputs and physical system setups. Integrating a new sensor or varying the sensor sets across different vehicle models requires major engineering effort.

These challenges hinder adoption of new sensors and create a perception software infrastructure that limits Late Fusion's applicability to higher levels of automated driving. With the increased

quality and performance demanded by automated driving systems, developing complex new software stacks and unique perception systems that meet requirements for each hardware setup could become more expensive than developing new sensors. The industry needs a perception approach that scales across sensors and vehicle models, creating additive capabilities for next-generation systems to continually improve performance by building on predecessors.

Deep Sensor Fusion

The alternative to Late Sensor Fusion is Early Sensor Fusion. Early Fusion is also sometimes called Raw Fusion or Low-level Fusion. At DeepScale, we call our development *Deep Sensor Fusion* (DSF), which is Early Fusion that is powered by Deep Learning. DSF uses Deep Neural Networks (DNNs) to create one holistic 3D representation of the environment based on information from all available sensors. This approach simplifies the perception system:

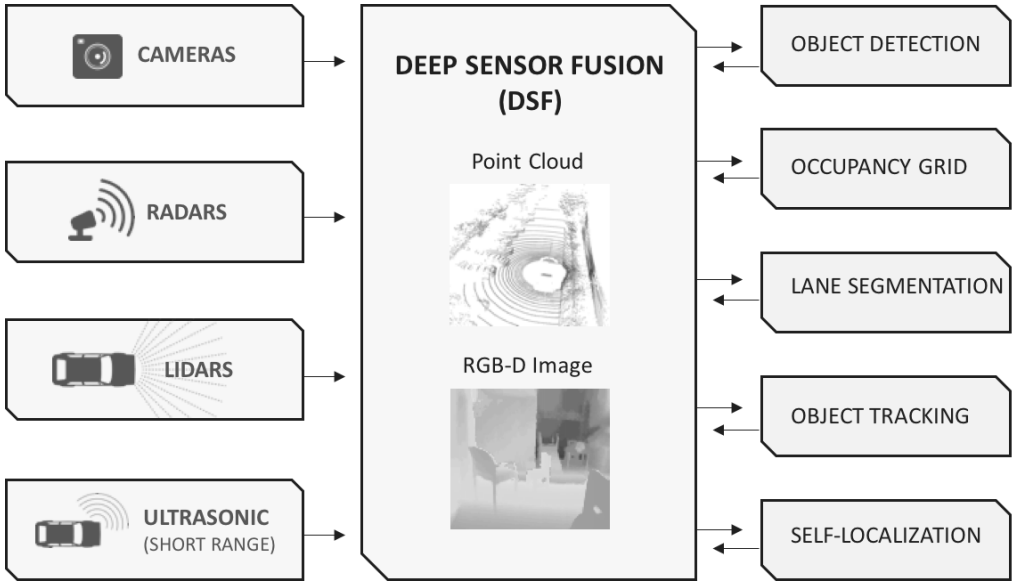

Figure 4: Deep Sensor Fusion Perception System

DSF is the foundation for improved perception because it creates a normalized representation of the environment that is sensor agnostic; regardless of what sensor feeds data to the DSF software, the output of this intermediate DSF model looks similar to the perception tasks (e.g., object detection and lane segmentation). DeepScale then also implements the perception tasks via Deep Learning. Thus, DSF introduces solutions to the shortcomings of Late Fusion:

1) DSF preserves detailed information about the data from each sensor before making perception predictions. Then, the perception predictions contain innate information that is important for troubleshooting. For example, DSF can assign confidence scores based on how much of an object was visible during a prediction or detect when a camera is contributing low-fidelity information to the model because of direct sunlight.
2) By fusing raw data, DSF creates a system that is greater than the sum of its parts. LIDAR doesn't get good returns from glass or low-reflectivity objects, camera performance degrades at night or during inclement weather, and today's RADARs have narrow

requirements for material compositions and velocities that register good returns. These problems are inherent to the physics of these sensors. With DSF, the strengths of each sensor supplement the weaknesses of other sensors to produce a better integrated model.

3) The DSF and perception software is similar across different automated driving implementations. Making a change in the sensor setup requires calibration of its interface to the DSF model—as opposed to re-writing a unique end-to-end stack of code for that sensor and system. Also, DSF enables sensor super-sets to quantify the performance of their sub-sets. This is a useful metric for advising an automated vehicle how to react to a sensor failure, and it is also an analogy to enabling lower-budget vehicle models to use sub-sets of high-end vehicle hardware while maintaining common software.

Ultimately, regardless of the type of approach used for perception, the software must run on an embedded processor in the vehicle since automated driving is a safety-critical application that cannot tolerate latency. Publicly known autonomous vehicle prototypes have high-power servers running in the trunk to handle all of the computation required. This is not commercially feasible in mass production. DeepScale focuses on running efficient DNNs on processor power and cost budgets that align with vehicle cost expectations [3]. DSF and its associated perception DNNs are modular and can be implemented across several different processors for various automated driving use cases [4].

Conclusions

The points presented above might force people to wonder why Late Fusion was ever used at all. Firstly, it's important to acknowledge that Late Fusion sufficed for solving lower-level ADAS functions that play an important role in accident prevention today. Furthermore, Late Fusion historically fit the technologies available in vehicles better than Early Fusion. For Early Fusion, insufficient computational capability and slow in-vehicle networking posed major barriers to entry. Thankfully, trends like mandated rear-view cameras and infotainment systems that race to keep up with consumer electronics have created the compute and bandwidth infrastructures required to enable Early Fusion and improve automated driving systems.

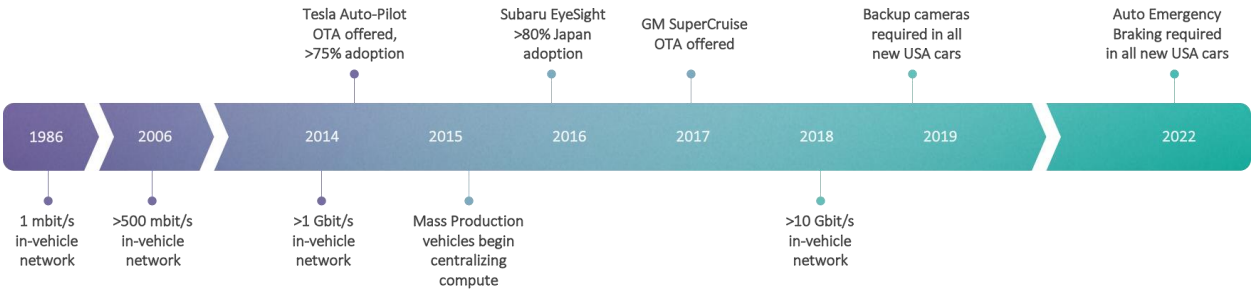

Figure 4: Trends Enabling Improved Perception [5-6]

Perhaps most importantly, there is strong market pull for ADAS and improved safety, which has resulted in sensors being installed in most new vehicles and trending towards coverage of all vehicles. These upgraded infrastructures and ubiquitous sensors combine with over-the-air updates to unbundle new driving features from hardware that could historically only be

purchased via expensive dealer upgrades that was forced to follow slow automotive development cycles. There is now a platform for software-defined vehicles that have the capacity to continue improving over time through data collection and software updates. Offerings like DeepScale's can create better performing ADAS and automated vehicles while unlocking system cost savings and creating new revenue streams for the automotive industry.

References

[1] http://www.sae.org/misc/pdfs/automated_driving.pdf

[2] https://www.theverge.com/platform/amp/2017/8/16/16155254/argo-ai-ford-self-driving-car-autonomous

[3] F.N. Iandola, M. Moskewicz, K. Ashraf, S. Han, W. Dally, K. Keutzer. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <1MB model size. arXiv, 2016.

[4] M.W. Moskewicz, F.N. Iandola, K. Keutzer. Boda-RTC: Productive Generation of Portable, Efficient Code for Convolutional Neural Networks on Mobile Computing Platforms. WiMob, 2016.

[5] http://www.autonews.com/article/20131007/OEM06/310079970/subaru-improves-safety-system

[6] https://en.wikipedia.org/wiki/Vehicle_bus