Sensing Optimization in Automotive Platforms

Joydeep Dey and Sudeep Pasricha

1 Introduction

The increasing maturity of Advanced Driver Assistance Systems (ADAS) [30] is enabling the introduction of vehicles with greater levels of autonomy. The degree to which ADAS can effectively reduce human intervention during driving is classified by SAE according to the J3016 standard [1], into five levels of autonomy.

Level 0 characterizes vehicles that have no assistive features. Level 1 autonomy encompasses vehicles that have the ability to share control between the driver and the vehicle. Adaptive cruise control and park assist are examples of features that can assist the driver in this level. Level 2 autonomy vehicles have the capability to perform all acceleration, steering, and braking tasks that require longitudinal and lateral control. Examples of features supported in this level include forward collision warning and blind spot warning, in addition to features from level 1. Level 3 autonomy vehicles can assess the risk of a situation and additionally perform path planning. At Level 4 autonomy, no driver intervention is required in most cases, unless requested, in contrast to level 3. Level 5 autonomy requires no human intervention or safety driver in the vehicle, unlike in level 4. *Most vehicles today are beginning to support level 2 autonomy.*

The higher autonomy levels require support for increasingly sophisticated ADAS features such as Lane Keep Assist (LKA) and Forward Collision Warning (FCW), which in turn defines requirements for sensing capabilities and perception

J. Dey (✉)
Department of Electrical and Computer Engineering, Colorado State University, Fort Collins, CO, USA
e-mail: Joydeep.Dey@colostate.edu

S. Pasricha
Colorado State University, Fort Collins, CO, USA
e-mail: sudeep@colostate.edu

V. K. Kukkala, S. Pasricha (eds.), *Machine Learning and Optimization Techniques for Automotive Cyber-Physical Systems*, https://doi.org/10.1007/978-3-031-28016-0_19

Table 1 ADAS sensor trade offs

Characteristics	Camera	LiDAR	Radar
Perception reliability	Medium	High	Medium
Spatial resolution	High	High	Low
Noise susceptibility	High	Low	Low
Velocity detection	Low	Low	High
Weather durability	Low	Low	High

performance of the vehicle. This increased demand for vehicle autonomy resulted in various challenges related to reliability [31–34], security [35–39], and real-time perception [40–44] of the vehicle. In this chapter, we focus on real-time perception, specifically, challenges associated with sensor configuration for achieving vehicle autonomy goals. Table 3 summarizes the trade-offs between popular sensors used to support ADAS features and their relative performance. Using a camera as a vision sensor is a widely used approach to perform the classification and detection of objects on the road. However, cameras have high susceptibility to noise and are not reliable in extreme weather or lighting conditions [2]. A radar sensor is also capable of object detection and is particularly suited for accurate velocity detection of neighboring vehicles even under harsh weather and poor visibility conditions. Long-range radars (typically at 77GHz) used to support ADAS features such as adaptive cruise control (ACC) and automatic emergency braking (AEB) have a shorter azimuth than mid or short-range radars (typically at 24GHz), to prioritize monitoring vehicle velocity and approaching distance. However, long range radars can also detect more number of objects than short or mid-range radars. A drawback of the radar is their high false positive rate when detecting objects, and an upper bound on the number of objects that can be detected at the same time, e.g., the Bosch midrange radar with a maximum range of 160 meters can only detect up to 32 objects simultaneously [3]. A LiDAR sensor uses invisible laser light to measure the distance to objects in a similar way to radars. It can create an incredibly detailed 3D view (point cloud) of the environment around the vehicle. However, LiDAR data processing is computationally very expensive and relies on moving parts which can make it more vulnerable to damage. Ultrasonic sensors listed in Table 3 use the principle of 'time of flight' to measure distance from targets by computing the travel time of the ultrasonic echo from a neighboring vehicle or obstacle [4]. Usage of ultrasonic sensors for ADAS feature implementation are not uncommon, however they require accurate modelling for their use case, since their performance is highly dependent on the physical properties (shape, surface material) of the target being tracked [5] (Table 1).

Most level 2 and higher autonomy vehicles today rely on a combination of sensors, to overcome their individual drawbacks (see Table 3). For example, Waymo (a subsidiary of Alphabet Inc., originally started as a project by Google in 2009) combines 3 different types of LiDAR sensors, 5 radar sensors, and 8 cameras. Tesla's vehicles avoid LiDARs due to their high costs and instead their Autopilot uses 8 surround cameras, 12 ultrasonic sensors (primarily for short-range self-parking support), and 1 forward-facing radar. Each of the cameras has a maximum

visibility range of up to 250 meters, so this configuration ensures a 360-degree coverage up to 250 meters around the vehicle.

An important challenge facing emerging vehicles is to determine a sensor configuration that can be responsible for environment perception as per the SAE autonomy level supported by the vehicle. An optimal sensor configuration should consist of carefully selected location and orientation of each sensor in a heterogeneous suite of sensors, to maximize coverage from the combined field of view obtained from the sensors, and also maintain a high object detection rate. Today there are no generalized rules for the synthesis of sensor configurations, as the location and orientation of sensors depends heavily on the target features and use cases to be supported in the vehicle.

In this chapter, we propose a novel framework called *VESPA* (VEhicle Sensor Placement and orientation for Autonomy) (first introduced in [43]), to optimize heterogeneous sensor synthesis. More precisely, for a given set of heterogeneous sensors and ADAS features to be supported, *VESPA* performs intelligent algorithmic design space exploration to determine the optimal placement and orientation for each sensor on the vehicle, to support the required ADAS features for SAE level 2 autonomy systems. The *VESPA* framework can be easily utilized to generate optimal sensor configurations across different vehicle types. Our experimental results indicate that the proposed framework is able to optimize perception performance across multiple ADAS features for the 2019 Chevrolet Blazer and 2016 Chevrolet Camaro vehicles.

2 Related Work

State-of-the-art SAE level 2 autonomy systems require the selection and placement of sensors based on the assistive target features required to be supported, e.g., forward collision warning (FCW) and lane keep assist (LKA). While several prior works evaluate the performance of a specific sensor configuration and its deployment, very few works have explored the problem of generating optimal sensor configurations for vehicles.

An optimal sensor placement approach was proposed in [6] for a blind spot detection and warning system. The work recognizes the inability of the camera to perform in non-ideal lighting conditions and selects an ultrasonic sensor to measure distance of vehicles trailing in the vehicle's blind spot. The time response of the system with the position of the sensor above the rear tire is analyzed for two scenarios: when the vehicle is at rest and when it is moving at a constant velocity. The sensor selection identifies price as a constraint and optimizes the price of the total sensor setup through usage of an ultrasonic sensor instead of a more expensive camera sensor. The work in [7] focuses on generating a LiDAR configuration from a set of LiDARs with the goal of reducing occurrences of dead zones and improving point cloud resolution. A LiDAR occupancy grid is constructed for a homogenous set of LiDARs and the configuration is generated using a genetic algorithm. An approach for optimal positioning and calibration of a three LiDAR

system is proposed in [8] that uses a neural network to qualify the effectiveness of different sensor location and orientations. Unlike these prior works that focus on generating configurations for a homogenous set of sensors, our work in this chapter presents a novel sensor placement and orientation optimization framework for a heterogeneous set of sensors. Moreover, our framework is also shown to be capable of easily adapting to different vehicle types.

3 Background

3.1 ADAS Features for Level 2 Autonomy

We target four ADAS features in this chapter that need to be supported by a deployed sensor configuration on a vehicle (henceforth referred to as an ego vehicle). A sensor configuration consists of the location and orientation of each sensor within a heterogeneous set of sensors. Our *VESPA* framework optimizes the sensor configuration to support four features: adaptive cruise control (ACC), lane keep assist (LKA) forward collision warning (FCW), and blind spot warning (BW). Each of the features discussed above, require varying degrees of sensing and control along longitudinal (i.e., within the same lane as the ego vehicle) and lateral (i.e., along neighboring lanes) regions.

SAE J3016 defines ACC and LKA individually as level 1 features, as they only perform the dynamic driving task in either the latitudinal or longitudinal direction of the vehicle. FCW and BW are defined in SAE J3016 as level 0 active safety systems, as they only enhance the performance of the driver without performing any portion of the dynamic driving task. However, when all four features are combined, the system can be described as a level 2 autonomy system. Many new vehicles being released today support level 2 autonomy. For instance, Volvo announced that its upcoming Level 2+ vehicles will use surround sensors for 360-degree perception, as well as deep neural networks running in parallel for robust object detection [9]. It is not only relevant, but also important to optimize sensor placement for ADAS systems as more and more vehicles with these features become available. Figure 1 shows an overview of the four features we focus on for level 2 autonomy, which are discussed next.

Adaptive cruise control (ACC) was first introduced in the Mercedes-Benz S-Class sedan in 1999, with the goal of increased driver comfort. ACC causes the ego vehicle to follow a lead vehicle at a specified distance (Fig. 1) without exceeding the speed limit specified by the operator upon activation of the feature [10]. If the lead vehicle slows down, then it is the responsibility of ACC to slow down the ego vehicle to maintain the specified distance. Although implementations differ, all ACC systems take over longitudinal control from the driver (Fig. 1). The challenge in ACC is to maintain an accurate track of the lead vehicle with a forward facing sensor and using longitudinal control to maintain the specified distance while maintaining driver comfort (e.g., avoiding sudden velocity changes).

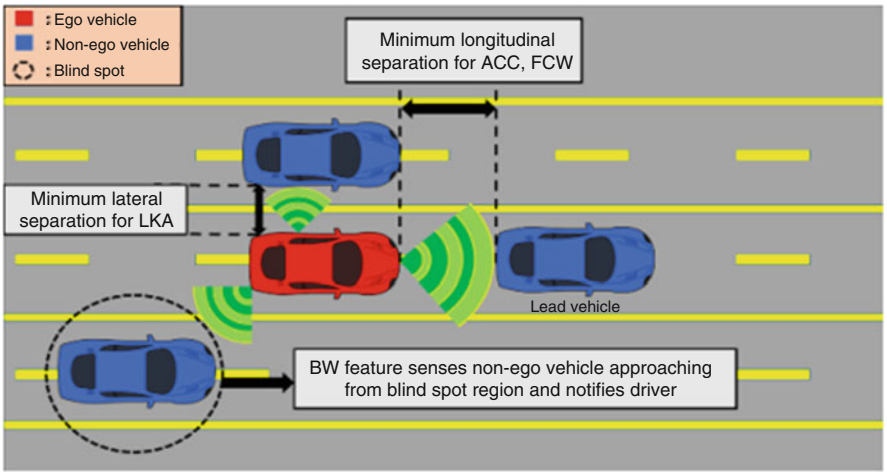


Fig. 1 Visualization of common scenarios in ACC, FCW, LKA, and BW

Lane keep assist (LKA) is an evolution of lane departure warning systems. It involves a forward-facing sensor (often a camera) to identify where the lane lines exist in front of the ego vehicle. Once the lane lines have been detected (e.g., using Canny edge detection and Hough transforms on forward-facing images), LKA can then determine if the ego vehicle lies between those lines (Fig. 1). If the ego vehicle appears to be drifting toward a position where it will cross lane line boundaries, LKA engages steering torque to steer the vehicle in the opposite direction of the lane line until it no longer has the trajectory to cross that lane. LKA systems have been known to over-compensate, creating a "ping-pong" effect where the vehicle oscillates back and forth between the lane lines [11]. The main challenges in LKA are to reduce this ping-pong effect and the accurate detection of lane lines on obscured (e.g., dirt covered) roads.

Forward collision warning (FCW) uses information gathered via various forward facing sensors to determine whether the ego vehicle is going to collide with an object in front of it (Fig. 1). As objects approach the boundary where the vehicle can no longer come to a stop, an audio-visual warning notifies drivers instructing them to apply the brakes. As this is a safety-critical system, it is important that FCW avoids false positives as well as false negatives to improve driver comfort, safety, and reduce rear end accidents [12]. For this to be achieved, it is a necessary prerequisite that the sensors used by the FCW system be placed where they have an accurate view of the vehicle in front of them. The United States National Transportation Safety Board has recommended that FCW be included in all new vehicles [13].

Lastly, blind spot warning (BW) uses sensors mounted on the sides of the ego vehicle to determine whether there is a vehicle towards the rear on either side of the ego vehicle in a location the driver cannot see with their side mirrors [14] (Fig. 1). This area is typically referred to as the "blind spot" and must be verified as

clear of any vehicles before the driver can attempt to make a lane change. Without BW, the driver must turn their head to make that verification on their own. With BW, the driver can maintain their concentration on the road ahead. As BW requires information about a specific area near the rear of the vehicle, it is a challenge to find an optimal sensor placement that maximizes the view of the blind spot. If the sensor is too far forward, it will miss the blind spots entirely, causing a vehicle accident when the driver makes a lane change. If the sensor is too far back, it will end up capturing information for areas around the ego vehicle that are not in the blind spot, decreasing the sensor's effectiveness at viewing the presence of vehicles surrounding the blind spot.

3.2 Feature Performance Metrics

To quantify the performance of a sensor configuration on a vehicle being evaluated over drive cycle test cases (i.e., across various driving scenarios; see Section V), we define eight metrics (m1–m8) that are characteristic of the configuration's ability to track and detect non-ego vehicles across various road geometries and traffic scenarios. The eight metrics are defined as follows:

$$\text{Longitudinal Position Error (m1)} = \frac{\sum (y - ygroundtruth)}{Number\ of\ non\ ego\ vehicle} \tag{1}$$

$$\text{Lateral Position Error (m2)} = \frac{\sum (x - xgroundtruth)}{Number\ of\ non\ ego\ vehicle} \tag{2}$$

$$\text{Object Occlusion Rate (m3)} = \frac{\text{Number of non ego vehicle undetected}}{Total\ number\ of\ passing\ non\ ego\ vehicles} \tag{3}$$

$$\text{Velocity Uncertainty (m4)} = \frac{\text{Number of invalid detected non ego vehicle velocities}}{Total\ number\ of\ non\ ego\ velocities} \tag{4}$$

$$\text{Rate of late detection (m5)} = \frac{\text{Number of late non ego vehicle detection}}{Total\ number\ of\ non\ ego\ vehicles} \tag{5}$$

$$\text{False positive lane detecion rate (m6)} = \frac{\text{Number of false positive lane detections}}{Total\ number\ of\ lane\ detections} \tag{6}$$

$$\text{False negative lane detecion rate (m7)} = \frac{\text{Number of false negative lane detections}}{Total\ number\ of\ lane\ detections} \tag{7}$$

$$\text{False positive object detecion rate (m8)} = \frac{\text{Number of false positive non ego vehicle detections}}{\text{Total number of non ego vehicle detections}} \tag{8}$$

The longitudinal position error *(m1)* and lateral position error *(m2)* are computed as the deviation of the positional data detected by the sensor configuration from the ground truth of non-ego vehicle positions along the y and x axes respectively. The lateral position error is relevant for LKA, while longitudinal position error is most relevant for ACC and FCW. The object occlusion rate *(m3)* measures the percentage of passing non-ego vehicles that go undetected in the vicinity of the ego vehicle. The minimization of this metric optimizes BW capabilities of a sensor configuration. The velocity uncertainty *(m4)* is the fraction of times that the velocity of a non-ego vehicle is measured incorrectly, which matters for ACC and FCW. The rate of late detection metric *(m5)* is computed as a fraction of the number of 'late' non ego vehicle detections made by the total number of non-ego vehicles, which matters for BW. A detection is classified as late if it is made after the non-ego vehicle crosses the minimum safe longitudinal or lateral distance defined by Intel RSS (Responsibility Sensitive Safety) models on NHTSA for pre-crash scenarios [15]. When a lane marker is detected but there exists no ground truth lane in simulation it is classified as a false positive lane detection, conversely, if a ground truth lane exists in simulation but is not detected, it is classified as a false negative lane detection [16]. Metrics 6 and 7 *(m6 and m7)* characterize the perception system's ability to make a correct case for lane keep assist by taking into account the false positive and false negative lane detection rate. False positive object detection rate *(m8)* measures the fraction of total vehicle detections which were classified as non-ego vehicle detections but did not actually exist in ground truth in the test cases.

4 *VESPA* Framework

The following section describes the proposed *VESPA* framework in detail.

4.1 Overview

Figure 2 shows an overview of our proposed *VESPA* framework. The physical dimensions of the vehicle model and the number and type of sensors to be considered are inputs to the framework. A design space exploration algorithm is

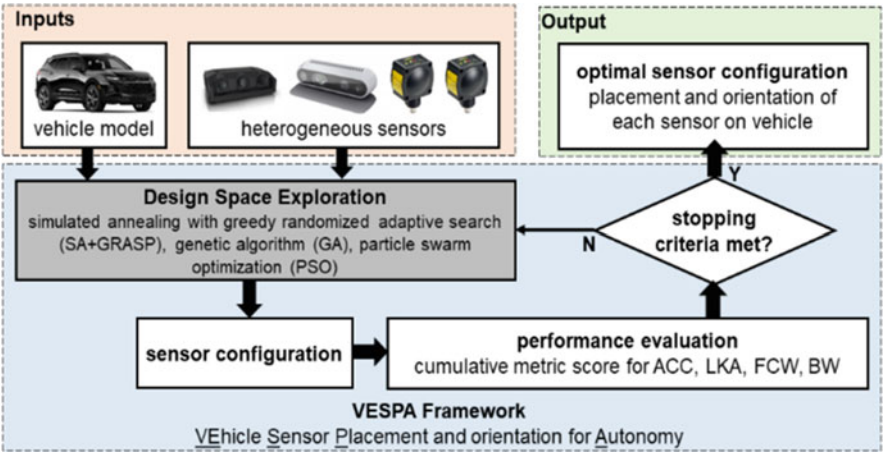


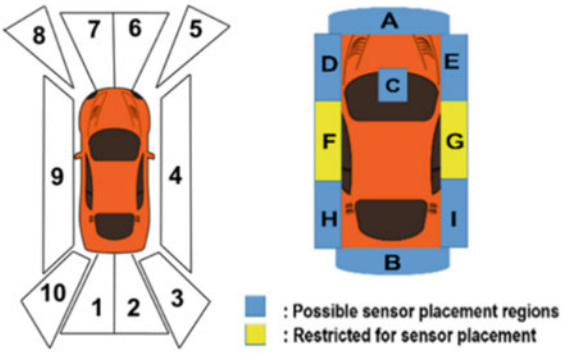
Fig. 2 Overview of *VESPA* framework

used to generate a sensor configuration which is subsequently evaluated based on a cumulative score from the performance metrics presented in the previous section. We evaluate three design space exploration algorithms: simulated annealing with greedy randomized adaptive search (SA + GRASP), genetic algorithm (GA), and particle swarm optimization (PSO). The process of sensor configuration generation and evaluation continues until an algorithm-specific stopping criteria is met, at which point the best configuration is output. The following subsections describe our framework in more detail.

4.2 Inputs

Each of the design space exploration algorithms generates sensor configurations that consider feature to field of view (FOV) zone correlations around the ego vehicle. Figure 3a shows the FOV zones around the ego-vehicle. These zones of interest are defined as the most important perception areas in the environment for a particular feature. Figure 3b shows the regions on the vehicle on which sensors can be mounted (in blue). Regions F and G (in yellow) are exempt from sensor placement due to the mechanical instability of placing sensors on the door of a vehicle.

The correlation between features, zones, regions, and performance metrics shown in Fig. 3 is summarized in Table 4. For example, in Fig. 3a, for ACC, the zones of interests are 6, and 7, and the corresponding regions for possible sensor placement are A and C. For exploration of possible locations within a region, a fixed step size of 5 cm in two dimensions across the surface of the vehicle is considered, which generates a 2D grid of possible positions in each zone shown in Fig. 3b, c. The orientation exploration of each sensor involves rotation at a fixed step size



Zone	Number of unique positions in each zone	Number of unique orientations at each position
A	400	5
B	400	5
C	200	5
D	200	10
E	200	10
H	400	10
I	400	10

(a) (b) (c)

Fig. 3 (**a**) Field of view (FOV) zones; (**b**) sensor placement regions; (**c**) design space breakdown

Table 2 Feature, region, zone and performance metric relationship

Feature	Region	Zone	Associated metrics
BW	B.H.I	1, 2,3,10	(m3, m5, m8)
LKA	E, I	3,4,5	(m2, m3, m6, m7)
ACC, FCW	D, H	8, 9, 10	(m1, m4, m8)
	A, B,C	6, 7, 11	

of 1 degree between an upper and lower bounding limit for roll, pitch and yaw respectively, at each of these possible positions within the 2D grid.

The orientation exploration limits were chosen with caution to the caveat that long range radars with extreme orientations increase the number of recorded false positives. The combined position and orientation exploration generates an intractably large design space as discussed next (Table 2).

4.3 Design Space Exploration

All of the metrics (m1 – m8) defined in 2.3.2 represent good performance at lower values. We create a cost function that combines these metrics and frame our sensor placement and optimization problem as a minimization problem. The most important metrics are identified and grouped for each feature, as shown in Table 4, and are used to model the cost function as a weighted sum of these five metrics, where the weights are chosen on the basis of their total cardinality across all feature. By searching through the design space of sensor configurations for a minimum cost function value, a sensor configuration can thus be generated where the metrics are cumulatively minimized.

The design space considered in this chapter uses 4 radars and 4 cameras that can be placed in any zone. With a fixed step size of 5 cm in each dimensions and 1 degree

rotation in orientation, the number of ways 8 sensors can be placed in all unique locations and orientations is 2.56e+23C8 for the 2019 Blazer and 6.4e+22C8 for the 2016 Camaro. As this design space is so large that it cannot be exhaustively traversed in a practical amount of time, we explore the use of intelligent design space search algorithms that support hill climbing to escape local minima. The three algorithms implemented as part of *VESPA* are discussed next.

4.3.1 SA + Greedy Random Adaptive Search Procedure (SA + Grasp)

Simulated annealing (SA) is a search algorithm that is useful in finding the global optima when the design space has multiple local optima [17]. The process is analogous to the way metals cool and anneal [18]. Typically, SA picks the best solution at each iteration, but can also pick the worst solution based on a temperature-dependent probability, which can allow it to climb out of local minima to arrive at global minima [19]. But SA suffers from the drawback of behaving like a greedy algorithm at lower temperatures as it tends to accepts only those solution configurations very close in cost function value to the previous solution, so it can get stuck in local minima in more complex design spaces [20]. The GRASP (Greedy Randomized Adaptive Search Procedure) algorithm is another search algorithm that is used in many exploration problems [21], but it does not always generate optimal solutions during the greedy construction phase and can get stuck in local optima easily. The SA + GRASP algorithm eliminates the inherent drawbacks of each algorithm. Specifically, the greedy randomized construction phase of the algorithm is used to create disturbances in the existing list of best sensor configurations in our problem, to generate better solutions. A new solution is generated in each iteration by selecting the better solution between the greedy solution from the greedy randomized construction phase and the configuration found from the local search. We decreased the SA temperature variable from Tmax = 10,000 to Tmin = 0 at the rate of 4 degrees per iteration. The search repeats by decreasing SA temperature till an optimal solution is found or a stopping criterion is achieved.

4.3.2 Genetic Algorithm (GA)

The GA is an evolutionary algorithm that can solve optimization problems by mimicking the process of natural selection [22]. It repeatedly selects a population of candidate solutions and then improves the solutions by modifying them. GA has the ability to optimize problems where the design space is discontinuous and also if the cost function is non differentiable [23]. The GA is adapted for our design space such that a chromosome is defined by the combined location and orientation of each sensor's configuration (consisting of six parameters: x, y, z, roll, pitch, and yaw). For a given set of N sensors, the number of parameters stored in each chromosomes is thus '6 N'. Next, in the selection stage, the cost function values are computed for 100 configurations at a time, and a roulette wheel selection method is used to

select which set of chromosomes will be involved in the crossover step based on their cost function probability value, computed as a fraction of the cumulative cost function sum of all chromosomes considered in the selection. In the crossover stage, the crossover parameter is set to 0.5, which allows 50 out of the 100 chromosomes to produce offspring. The mutation parameter is set to 0.2 such that in the mutation stage, the mutation rate is set to 10, which is the number of new genes allowed for mutation in each iteration.

4.3.3 Particle Swarm Optimization (PSO)

PSO considers a group of particles where each particle has a position and velocity and is a solution to the optimization problem [24]. In our problem each sensor configuration in the design space is represented as a particle having a defined position and velocity. With a random start, the cost function in (5) evaluates the quality of the solution of a particle. The particle's velocity and position values are updated recursively using a linear update [24]. Each particle stores a trace of its best position within the group and globally as well. The history of the cost function values for this trace can explain the effectiveness of changing the position of a particular sensor from the set of heterogeneous sensors [25]. Unlike GA, PSO does not have any evolution operators like crossovers or mutation [26]. PSO also does not require any binary encoding of solution configurations like in GA [27]. The total number of particles considered were 50, and the importance of personal best and importance of neighborhood best parameters were both empirically selected to be 2.

5 Experiments

The following section describes the experimental setup and results involving the *VESPA* framework.

5.1 Experimental Setup

To evaluate our *VESPA* framework, we consider a scenario with a maximum of 8 sensors: 4 radars and 4 camera vision sensors. Many recent contributions such as the work presented in [28, 29] combine radar and camera modalities for ADAS applications. We did not include LiDARs in this heterogeneous set of sensors due to their relatively poor performance in adverse weather conditions as shown in Table 3. For the given set of test cases, it was observed that if less than 4 sensors were used, the ability of the perception system to make an accurate prediction was relatively poor. Conversely, on increasing the number of radars and cameras to more than

Table 3 *VESPA* generated solution vs baseline configuration

	VESPA Camaro	Baseline Camaro	*VESPA* Blazer	Baseline Blazer
Cost function	0.9971	2.1367	1.2841	2.4630
Longitudinal position error	0.0523	0.1427	0.0845	0.2419
Lateral position error	0.1810	0.2566	0.0958	0.2204
Object occlusion rate	0.1331	0.2351	0.2062	0.3158
Velocity uncertainty	0.0823	0.1851	0.0474	0.2056
Rate of late detection	0.1158	0.2123	0.1578	0.2315
False positive lane detection rate	0.0142	0.1335	0.0221	0.1571
False negative lane detection rate	0.0214	0.0236	0.0393	0.0412
False positive object detection rate	0.0431	0.1283	0.0976	0.0954

	Length	Width	Height	Front Overhang	Rear Overhang
Blazer	4.7498	1.9558	1.6764	1.016	0.9144
Camaro	4.7498	1.905	1.3716	0.9652	1.016

Fig. 4 2019 Chevrolet Blazer (Left) and 2016 Chevrolet Camaro (Right)

4 each, there was minimal improvement in cost function score. Hence to keep implementation cost low while still achieving good accuracy, we decided to use these 8 sensors. Please note that these modalities and number of sensors have been used to show a proof of concept for our *VESPA* framework, which can be extended to scenarios with different modalities and numbers of sensors. We considered two vehicles for evaluation: a 2019 Chevrolet Blazer and a 2016 Chevrolet Camaro. Figure 4 shows the dimensions for the vehicles. Figure 5 shows images of the sensor placements on both car models in our workspace.

Each configuration generated by the SA + GRASP, GA, and PSO algorithms was optimized on 40 test cases designed (10 test cases each for evaluating performance with ACC, FCW, LKA, and BW) using the Automated Driving Toolbox in Matlab. Half (20) of these test cases for each feature are used during the optimization phase and the remaining (20) test cases are used during the evaluation phase.

Fig. 5 Sensors mounted in workspace on both car models

Finally, the optimized configurations were evaluated on a different set of evaluation test cases. Each of the test cases was characterized by unique road geometry, variations in road elevation, curvature, banking, and different traffic densities. In some test cases, the number of lanes were varied to make the framework optimize the sensor configuration for challenging and realistic driving scenarios.

A Kalman filter sensor fusion algorithm was used to combine readings from sensors in a sensor configuration being evaluated, to make predictions. The longitudinal and lateral ground truth were defined for non-ego vehicles and the position error was calculated from the fused sensor measurements. The deviation of sensor measurements from ground-truth was used to calculate the values of metrics m1–m8, and hence the cost function over all test cases. Lastly, we set the stopping criterion for all three algorithms as the case when the cost function does not show a greater than 5% change over 200 iterations.

5.2 Experimental Results

In our first experiment we were interested in evaluating the efficacy of different optimization algorithms (SA + GRASP, GA, and PSO) in finding optimal sensor configurations as well as exploring the consistency of the quality of solution returned by each. The cost function values for the best solution found by each algorithm for the 2016 Camaro and 2019 Blazer are shown in Fig. 6 As shown in Fig. 6, GA returned the solution configuration with the lowest cost function score of 0.7648 for the Camaro and 0.9252 for the Blazer. GA was able to better traverse the complex design space for our problem to arrive at the global minima compared to the SA + GRASP and PSO algorithms.

Next, we compared the solution generated by *VESPA* (utilizing the GA algorithm which gives the best results) with a baseline sensor configuration selected manually, based on best practices by a vehicle design expert in our team. This baseline configuration involved coupling a radar and camera in zones A, B, E and H each such that every mutually perpendicular direction in the 2D plane of the ego vehicle was covered using a radar and camera combined. All 8 sensors were fixed in the orientation angle, which matched the orientation of surface normal vector of the

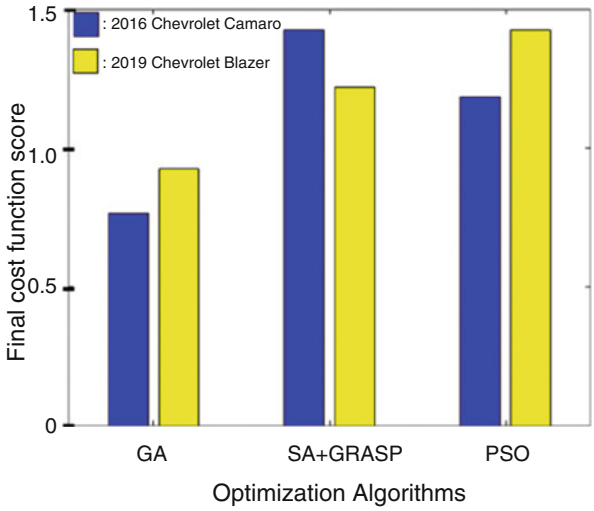


Fig. 6 Cost function values for the best solution found by the SA + GRASP, GA, and PSO algorithms on the Camaro and Blazer vehicles

respective zone in which they were placed. The selected baseline configuration maximizes coverage by considering feature to zone correlation. This is ensured by placing at least one sensor in each region such that all zones dedicated to each of the 4 selected features is covered in the field of view of that particular sensor.

Table 3 shows the results of the comparison between the *VESPA* generated solution and the baseline configuration for the 2016 Camaro and 2019 Blazer. The final cost function score was higher for the baseline approach, showing that *VESPA* generated a significantly better (lower cost) solution for both vehicles.

Table 4 summarizes the specific locations and orientations of the eight sensors on the two vehicles, generated by *VESPA*. The location and orientation information of each sensor in Table 6 is measured with respect to a global co-ordinate frame for the car model, whose origin is at the geometric center of the vehicle. An interesting observation from the table is that the sensors in the Blazer's configuration favor higher Z values than the Camaro, since the Blazer is 0.3 m taller than the Camaro.

Figure 7 visualizes sensor coverage in a bird's eye plot between the best configuration generated by *VESPA* in Fig. 7a and the baseline configuration in Fig. 7b for the Camaro (results for Blazer are omitted for brevity). The baseline configuration was optimized with a conventional approach towards improving sensor coverage, with a secondary focus on sensor reliability.

In contrast, the solution generated by *VESPA* took into account the unique strengths and weaknesses of each sensor to obtain a configuration having significantly better performance for the features supported, despite having lower overlap between field of view of different sensors than the baseline solution Fig. 7 and also uses lesser number of sensors. The superiority of the *VESPA* solution configuration, despite using lesser number of sensors, can be accounted for by the optimized placement of camera 1, radar 2 and radar 3 in zones A and C maximizing performance of ACC and FCW. Further, in physical testing it was observed that

Table 4 Solution from *VESPA* for Camaro and Blazer (in meters, degrees)

	Radar 1		**Radar 2**		**Radar 3**		**Radar 4**	
	VESPA camaro	*VESPA* blazer	*VESPA* camaro	*VESPA* blazer	*VESPA* camaro	*VESPA* blazer	*VESPA* camaro	*VESPA* blazer
X	3.7	3.7	3.7	3.65	0	2.8	3.7	2.91
Y	−0.18	−0.4	0.45	−0.9	0.9	0.9	−0.9	0.9
Z	0.15	0.2	0.20	0.2	0.2	0.25	0.2	0.2
Roll	0	0	0	0	0	0	0	0
Pitch	0	0	0	0	0	0	0	0
Yaw	15	−20	−15	−40	50	45	−130	−130
	Camera 1		**Camera 2**		**Camera 3**		**Camera 4**	
	VESPA camaro	*VESPA* blazer	*VESPA* camaro	*VESPA* blazer	*VESPA* camaro	*VESPA* blazer	*VESPA* camaro	*VESPA* blazer
X	3.7	3.7	2.8	2.8	2.8	0	X	−1
Y	0	0	−0.9	−0.9	0.9	0.9	X	0.9
Z	1.1	1.1	1.1	1.15	1.1	1.25	X	1.1
Roll	0	0	0	0	1	0	X	0
Pitch	0	0	0	0	1	1	X	1
Yaw	0	0	−90	−100	120	60	X	170

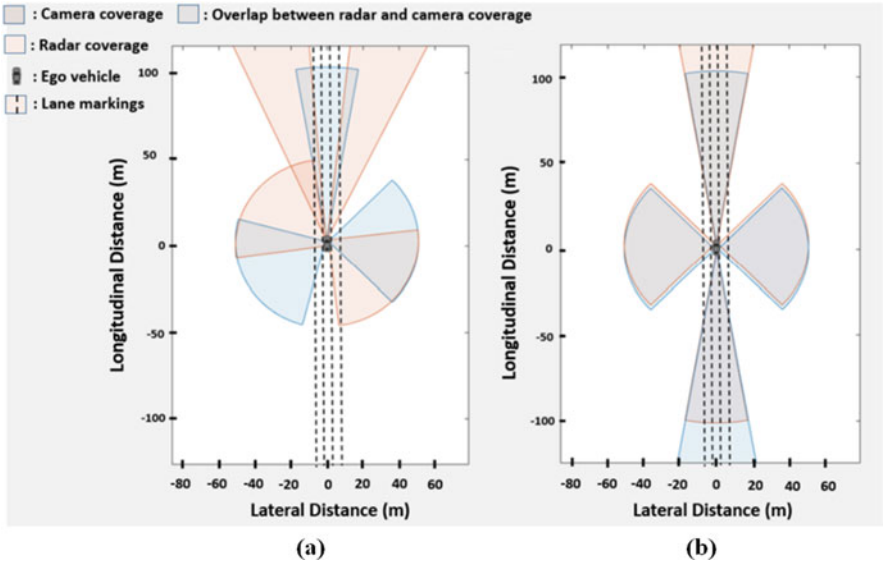


Fig. 7 Coverage for (**a**): *VESPA* Camaro solution (**b**) Baseline Camaro

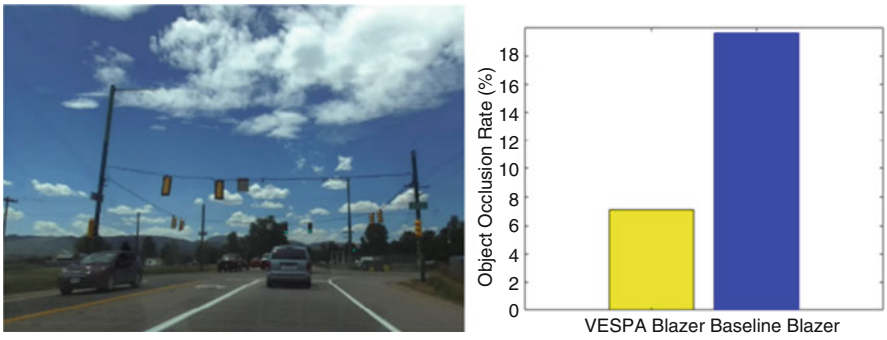


Fig. 8 Performance on real drive cycle in Colorado for best solution generated by *VESPA* and the baseline configuration for the 2019 Blazer

using a radar coupled with camera in zone B for LKA reduces the number of false positives during detections. In Fig. 7a, radars 3, 4 and cameras 2, 3 placed in zones D and E respectively were sufficient for improving performance of ACC and FCW by reducing the number of false positive object detections. The combined optimization of orientation and location with *VESPA* resulted in a sensor configuration that maximized performance for each feature.

Our last experiment involved testing the best sensor configuration from our *VESPA* framework and the baseline configuration for the 2019 Blazer on data from a real world drive cycle over 1 h in Colorado. We focus only on assessing performance for the ACC and FCW features. Figure 8 shows an image from the real drive cycle

with data collected by the vehicle from the radar and camera sensors on it. The figure also shows a plot of the object occlusion rate (OOR). The OOR for the baseline configuration was 19.64% (it did not detect 11 out of 56 non ego vehicles), while the *VESPA* generated best solution had an OOR of 7.14% (it failed to detect only 4 out of 56 non ego vehicles). The results show the effectiveness of our proposed *VESPA* framework in generating higher quality sensor configurations.

6 Conclusions

In this chapter, we propose an automated framework called *VESPA* that is capable of generating sensor placement and orientation in modern semi-autonomous vehicles. *VESPA* has the ability to optimize locations and orientations for a set of heterogeneous sensors on a given target vehicle. The framework can be tuned to improve perception on a desired collection of test cases. *VESPA* is also scalable across different vehicle models as shown in our analysis on the Chevrolet Camaro and Blazer vehicles. Further, despite the sensor locations in the baseline configuration of Fig. 7b being the most intuitive, the best configuration is the one generated by *VESPA*, showing that even people skilled in the art of sensor placement may find it challenging to synthesize a significantly better placement than that generated by *VESPA*. We also validated *VESPA* with real drive cycle data to show its effectiveness for real-world scenarios.

References

1. SAE International Standard J3016: Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles, (2018)
2. Spinneker, R., Koch, C., Park, S.B, Yoon, J.J: Fast fog detection for camera based advanced driver assistance systems. In: IEEE International Conference on Intelligent Transportation Systems (ITSC) (2014)
3. Robert Bosch GmbH, Bosch MRR Data Sheet: [Online]. Available: https://www.bosch-mobility-solutions.com, (2019)
4. Xu, C., Zhang, P., Wang, H., Li, Y., Li, C.: Ultrasonic echo waveshape features extraction based on QPSO-matching pursuit for online wear debris discrimination. Mech. Syst. Signal Process. **60**, 301–315 (2015)
5. Li, S., Li, G., Yu, J., Liu, C., Cheng, B., Wang, J., Li, K.: Kalman filter based tracking of moving objects using linear ultrasonic sensor array for road vehicles. Mech. Syst. Signal Process. **60**, 301–315 (2018)
6. Jamaluddin, M., Shukor, A.Z., Miskon, M.F., Ibrahim, F.A., Redzuan, M.Q.A.: An analysis of sensor placement for vehicle's blind spot detection and warning system. J. Telecommun. Electron. Comput. Eng. **8**(7), 101–106 (2017)
7. Kim, T., Park, T.: Placement optimization of multiple Lidar sensors for autonomous vehicles. IEEE Trans. Intell. Transp. Syst. **21**(5), 2139–2145 (2019)
8. Meadows, W., Hudson, C., Goodin, C., Dabbiru, L., Powell, B., Doude, M., Carruth, D., Islam, M., Ball, J.E, Tang, B.: Multi-LIDAR placement, calibration, co-registration, and processing on a Subaru Forester for off-road autonomous vehicles operations. In: Autonomous Systems: Sensors, Processing and Security for Vehicles and Infrastructure (2019)

9. Shapiro, D.: Levelling up: what is level 2 automated driving? [Online]. Available: https://blogs.nvidia.com/blog/2019/02/06/what-is-level-2-automateddriving/, (2019)
10. Wenger, J.: Automotive radar – status and perspectives. In: IEEE Compound Semiconductor Integrated Circuit Symposium (2005)
11. Kirchner, C.: Lane keeping assist explained. In: Motor Review, [Online]. Available: https://motorreview.com/lane-keeping-assist-explained, (2014)
12. Consumer Reports: Guide to Forward Collision Warning, Consumer Reports, [Online]. Available: https://www.consumerreports.org/car-safety/forwardcollision-warning-guide/ (2019)
13. National Transportation Safety Board (NTSB): The use of forward collision avoidance systems to prevent and mitigate rear-end crashes (2015)
14. Consumer Reports: Guide to blind spot warning, consumer reports, [Online]. Available: https://www.consumerreports.org/car-safety/blind-spotwarning-guide/, (2019)
15. Mobileye, Intel Co.: Implementing RSS Model on NHTSA Pre-Crash scenarios, [Online]. Available: https://www.mobileye.com/responsibility-sensitivesafety/rss_on_nhtsa.pdf, (2019)
16. Kumar, A., Simon, P.: Review of lane detection and tracking algorithms in advanced driver assistance systems. Int. J. Comput. Sci. Inf. Technol. **7**(4), 65–78 (2015)
17. Aarts, E., Korst, J.: Simulated annealing and Boltzmann machines. In: A Stochastic Approach to Combinatorial Optimization and Neural Computing. Wiley, New York (1989)
18. Bohachevsky, O., Johnson, M.E., Stein, M.L.: Generalized simulated annealing for function optimization. Technometrics. **28**(3), 209–217 (1986)
19. Van Laarhoven, P.J.M., Aarts, E.H.L.: Simulated Annealing: Theory and Applications Mathematics and Its Applications. Springer, Dordrecht (1987)
20. Oliveira, E., Antunes, C.H., Gomes, Á.: A hybrid multi-objective GRASP+SA algorithm with incorporation of preferences. In: IEEE Symposium on Computational Intelligence in Multi-Criteria DecisionMaking (MCDM) (2014)
21. Sosnowska, D.: Optimization of a simplified fleet assignment problem with metaheuristics: simulated annealing and GRASP. In: Nonconvex Optimization and Its Applications, Springer (2000)
22. Reeves, C.: Genetic algorithms: handbook of metaheuristics, In: International Series in Operations Research & Management Science, Springer (2003)
23. Rattray, M., Shapiro, J.L.: The dynamics of a genetic algorithm for a simple learning problem. J. Phys. A Math. Gen. **29**(23), 7451 (1996)
24. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of International Conference on Neural Networks (ICCN) (1995)
25. Clerc, M., Kennedy, J.: The particle swarm – explosion, stability, and convergence in a multidimensional complex space. In: IEEE Press (2002)
26. Huang, S.: A review of particle swarm optimization algorithm. In: Computer Engineering Design (1977)
27. Lei, X., Shi, Z.: Application and parameter analysis of particle swarm optimization algorithm in function optimization. In: Computer Engineering Applications (CEA) (2008)
28. Kumar, R., Jayashankar, S.: Radar and camera sensor fusion with ROS for autonomous driving. In: Fifth International Conference on Image Information Processing (ICIIP) (2019)
29. Zhong, Z., Liu, S., Matthew, M., Dubey, A.: Camera radar fusion for increased reliability in ADAS applications. In: Electronic Imaging, Autonomous Vehicles and Machines (2018)
30. Kukkala, V., Tunnell, J., Pasricha, S.: Advanced driver assistance systems: a path toward autonomous vehicles. In: IEEE Consumer Electronics, Vol. 7, No. 5 2018
31. Kukkala, V.K., Bradley, T., Pasricha, S.: Priority-based multi-level monitoring of signal integrity in a distributed powertrain control system. In Proceeding of IFAC Workshop on Engine and Powertrain Control, Simulation and Modeling (2015)
32. Kukkala, V.K., Bradley, T., Pasricha, S.: Uncertainty analysis and propagation for an auxiliary power module. In Proceeding of IEEE Transportation Electrification Conference (TEC) (2017)
33. Kukkala, V.K., Pasricha, S., Bradley, T.: JAMS: Jitter-aware message scheduling for FlexRay automotive networks. In: IEEE/ACM International Symposium on Networks-on-Chip (NOCS) (2017)

34. Kukkala, V.K., Pasricha, S., Bradley, T.: JAMS-SG: A Framework for Jitter-Aware Message Scheduling for Time-Triggered Automotive Networks. In: ACM Transactions on Design Automation of Electronic Systems (TODAES), Vol. 24, No. 6 (2019)
35. Kukkala, V.K., Pasricha, S., Bradley, T.: SEDAN: Security-aware design of time-critical automotive networks. In: IEEE Transaction on Vehicular Technology (TVT), Vol. 69, No. 8 (2020)
36. Kukkala, V.K., Thiruloga, S.V., Pasricha, S.: INDRA: intrusion detection using recurrent autoencoders in automotive embedded systems. In: IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD), Vol. 39, No. 11 (2020)
37. Kukkala, V.K., Thiruloga, S.V., Pasricha, S.: LATTE: LSTM Self-Attention based Anomaly Detection in Embedded Automotive Platforms. In: ACM Transactions on Embedded Computing Systems (TECS), Vol. 20, No. 5s, Article 67 (2021)
38. Thiruloga, S.V., Kukkala, V.K., Pasricha, S.: TENET: temporal CNN with attention for anomaly detection in automotive cyber-physical systems. In: Proceeding of IEEE/ACM Asia & South Pacific Design Automation Conference (ASPDAC) (2022)
39. Kukkala, V.K., Thiruloga, S.V., Pasricha, S.: Roadmap for cybersecurity in autonomous vehicles. In: IEEE Consumer Electronics Magazine (CEM), (2022)
40. Tunnell, J., Asher, Z., Pasricha, S., Bradley, T.H.: Towards improving vehicle fuel economy with ADAS. In: SAE Inter-national Journal of Connected and Automated Vehicles, Vol. 1, No. 2 (2018)
41. Tunnell, J., Asher, Z., Pasricha, S., Bradley, T.H.: Towards improving vehicle fuel economy with ADAS. In: Proceeding of SAE World Congress Experience (WCX) (2018)
42. Asher, Z., Tunnell, J., Baker, D.A., Fitzgerald, R.J., Banaei-Kashani, F., Pasricha, S., Bradley, T.H.: Enabling prediction for optimal fuel economy vehicle control. In: Proceeding of SAE World Congress Experience (WCX) (2018)
43. Dey, J., Taylor, W., Pasricha, S.: VESPA: a framework for optimizing heterogeneous sensor placement and orientation for autonomous vehicles. In: IEEE Consumer Electronics Magazine (CEM), Vol. 10, No. 2 (2021)
44. DiDomenico, G.C., Bair, J., Kukkala, V.K., Tunnell, J., Peyfuss, M., Kraus, M., Ax, J., Lazarri, J., Munin, M., Cooke, C., Christensen, E.: Colorado state university EcoCAR 3 final technical report. In: SAE World Congress Experience (WCX) (2019)