Augmented Informative Cooperative Perception

Pengyuan Zhou*, Pranvera Kortoçi*, Yui-Pan Yau†, Tristan Braud†, Xiujun Wang‡, Benjamin Finley*,
Lik-Hang Lee§, Sasu Tarkoma*, Jussi Kangasharju*, Pan Hui*

*University of Helsinki   †The Hong Kong University of Science and Technology
‡Anhui University of Technology   §University of Oulu

Emails: {firstname.lastname}@helsinki.com, {arthur.yau, lhleeac}@connect.ust.hk, braudt@ust.hk, wxj@mail.ustc.edu.cn

Abstract—Connected vehicles, whether equipped with advanced driver-assistance systems or fully autonomous, are currently constrained to visual information in their lines-of-sight. A cooperative perception system among vehicles increases their situational awareness by extending their perception ranges. Existing solutions imply significant network and computation load, as well as high flow of not-always-relevant data received by vehicles. To address such issues, and thus account for the inherently diverse informativeness of the data, we present Augmented Informative Cooperative Perception (AICP) as the first fast-filtering system which optimizes the informativeness of shared data at vehicles. AICP displays the filtered data to the drivers in augmented reality head-up display.

To this end, an informativeness maximization problem is presented for vehicles to select a subset of data to display to their drivers. Specifically, we propose (i) a dedicated system design with custom data structure and light-weight routing protocol for convenient data encapsulation, fast interpretation and transmission, and (ii) a comprehensive problem formulation and efficient fitness-based sorting algorithm to select the most valuable data to display at the application layer. We implement a proof-of-concept prototype of AICP with a bandwidth-hungry, latency-constrained real-life augmented reality application. The prototype realizes the informative-optimized cooperative perception with only 12.6 milliseconds additional latency. Next, we test the networking performance of AICP at scale and show that AICP effectively filter out less relevant packets and decreases the channel busy time.

I. Introduction

Connected and autonomous vehicles are closer than ever to becoming a reality. Specifically, modern communication technologies such as cellular vehicle-to-everything (C-V2X) and dedicated short-range communications (DSRC) facilitate large-scale vehicular communication thanks to significant improvements in bandwidth, latency, and reliability. Additionally, novel regulations provide a beneficial legal context for the operation of autonomous vehicles on public roads [1]. This paves the way for the deployment of applications that leverage vehicular communication to provide more information to human and AI drivers, and thus improve road safety. Currently, autonomous vehicles and advanced driver-assistance systems (ADAS) rely heavily on on-board sensors to identify and evaluate potential dangers and take necessary actions. However, most current solutions are limited to a single vehicle point of view, sensing only the nearby objects within their line-of-sight. As such, the vehicle's sensing capabilities are regularly obstructed by other vehicles, and thus depriving the driver of potentially useful information. Leveraging current and future
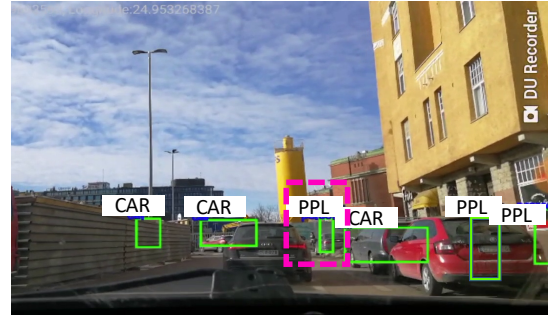

Fig. 1: Illustration of a naïve cooperative perception system. The driver of a following vehicle is shown all the objects detected by a leading vehicle. In contrast, AICP filters these objects to only show the critical objects (e.g., the pedestrians in the pink bounding box) to avoid information overload.

communication networks, the vehicle can aggregate the perception of multiple nearby vehicles, i.e., cooperative (collective) perception [2], [3], and provide a driver (human or AI) with a holistic view of the road situation. This concept has been adopted by the European Telecommunications Standards Institute (ETSI), which is working on Cooperative/Collective Perception Service standardization [4], [5].

Existing works focus on timely and synchronized information distribution, data fusion, or communication overhead [2], [3], [6], [7]. However, the informativeness of the shared perception data has been largely overlooked. Cooperative perception, and more generally safety applications that rely on communication among vehicles, require deployment at scale to provide a holistic vision of the road. Such a pervasive deployment leads to a significant strain in terms of network, computation resources, and driver awareness caused by the constant information dissemination across a large number of vehicles. However, only a part of the disseminated information is of interest to the drivers.

Figure 1 shows an example of a following vehicle's vision in a leading-following vehicle scenario with naïve cooperative perception with augmented reality (AR). The leading vehicle captures the objects within its line-of-sight and broadcasts information about the detected objects. The following vehicle calculates the position transformations and renders all the objects, shown in green boxes, from the received messages. Extending such a system to *city-block-level* perception with no information filtering leads to a massive number of objects being displayed to the drivers. This overwhelms their vision

and thus negatively impacts their driving experience. In fact, a driver's decision time increases logarithmically with the number of stimulus or objects [8]. Additionally, limiting the number of objects to the human cognition capacity of about 7 ± 2 items [9] is important. Therefore, cooperative perception requires an efficient filtering system. For instance, such a system might select the pedestrian, shown in the pink box in Figure 1, as highly-important and display their information while safely discarding the remaining objects.

The ETSI standards and a few recent works [7], [10] have proposed high-level descriptions of potential filtering rules and mechanisms. However, the topics of dedicated system design and fine-grained protocols that provide efficient data flow with light-weight operations and fast filtering algorithms that optimize the informativeness of objects in real-time have not been explored. In this paper, we propose Augmented Informative Cooperative Perception (AICP), the first solution that focuses on optimizing informativeness for pervasive cooperative perception systems with efficient filtering at both the network and application layers. AICP identifies, transmits, forwards, and filters objects at scale and displays the most important ones to the drivers through AR. Specifically, we make several key contributions as follows.

(1) We propose a system design for AICP. The design includes a dedicated data structure, the vehicular data unit (VDU), designed for informativeness-focused information filtering and transmission. We also describe the full-stack networking protocol that the system employs to utilize VDU.

(2) We formulate the informativeness problem in cooperative perception systems from the *object*, *message*, and *vehicle level*, and propose a weighted fitness sorting algorithm based on Mahalanobis distance for fast, yet comprehensive informativeness-based filtering. The algorithm provides filtering at the application level to display only the most important information shared by nearby vehicles, and thus preventing drivers from information overload.

(3) We implement a prototype of the proposal using a cooperative perception application on an Augmented Reality Head-up Display (ARHUD). Next, we evaluate the networking performance of AICP at scale with simulations.

We note that AICP is network-agnostic and does not depend on any particular feature of the underlying network, as such the system can be seamlessly integrated into current and future communication systems such as C-V2X and DSRC.

The rest of the paper is structured as follows. Section II discusses related works and states the key motivations behind AICP. Section III details the system architecture, data structure, and routing protocol. Section IV models the system and formulates the problem. Sections V describes the weighted fitness sorting algorithm to calibrate the assessment of informativeness. Section VI shows the proof-of-concept implementation of AICP and its performance in different scenarios. Section VII presents the simulation setup and results. Finally, Section VIII discusses system limitations and potential solutions, and Section IX concludes the work.

II. RELATED WORK

As this work touches on different research areas including cooperative perception, information filtering, and AR in the context of vehicular networks, the related work also falls into these areas. In terms of general vehicular cooperative perception, Kim et al. proposed a framework addressing several important problems in the field such as map merging, communication uncertainty, and sensor multi-modality [11]. Furthermore, [12] proposed and analyzed different message formats based on ETSI ITS 5G [13] to exchange local sensory data among road participants for collective perception. Thereafter, they proposed a multimodal cooperative perception system with a focus on engineering feasibility [3], and generalized the work with a mirror neuron inspired intention awareness algorithm for cooperative autonomous driving [14].

The critical area of filtering mechanisms in vehicular cooperative perception is still very new and has only a few key works. Garlichs et al. [7] in 2019 suggested a set of generation rules to reduce the transmission load while guaranteeing perception capabilities. This proposal was later added to the ETSI standard [5]. Thandavarayan et al. studied the ETSI standards and conducted an in-depth evaluation of the message generation rules [10]. They investigated the trade-off between the perception capabilities and communication performance under current standards and concluded that further optimization is needed to reduce information redundancy. To this end, Aoki et al. [15] applied a deep reinforcement learning approach to reduce the information sent between vehicles by only forwarding information about objects that are not likely to have been seen directly by surrounding vehicles themselves. Finally, works in the area of ARHUD, an in-car deployment of AR that visualizes information in the driver's line-of-sight, are also related. For instance, [6], [16] explore how to share augmented vision between two vehicles. Other studies consider connecting multiple mobile points of view to recompose a scene in 2D or 3D [17]. However, these studies mainly focus on image stitching, and thus overlooking aspects such as that of data redundancy.

To the best of our knowledge, cooperative augmented vehicular vision at scale still requires additional research on efficient information filtering. Specifically, we believe that AR-powered cooperative perception needs a comprehensive solution to maximize the informativeness of the data shared among vehicles to improve the driving experience while increasing road safety. Therefore, in this work we propose AICP, a system that lessens the burden on the network through efficient data filtering. Consequently, only the most relevant data is broadcast to vehicles, which in turn sort such data to maximize the informativeness that they yield to the driver.

Overall, we find that AR-powered cooperative perception needs a comprehensive solution to maximize the informativeness for better driver experience. In this paper, we propose AICP with a detailed system modelling, protocol design and algorithms.
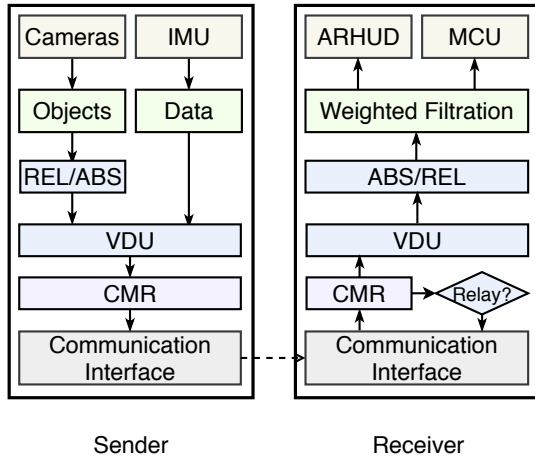
Fig. 2: System components: inertial measurement unit (IMU), microprocessor control unit (MCU), relative values (REL), absolute values (ABS), vehicular data unit (VDU), and contextual multihop routing (CMR) protocol.

Fig. 3: AICP protocol stack.

III. SYSTEM OVERVIEW

This section describes the proposed system in terms of the major components, data structure, and the routing protocol.

A. System Architecture

We consider a connected system of vehicles equipped with sensors and wireless communication modules that can collect and share sensory data with each other. We assume the system has capabilities including accurate positioning and localization [18], [19], relative velocity estimation, distance and angle estimation [20], and perspective transformation [6]. In this work we skip the details of the aforementioned techniques and focus on the aspect of information filtering. Figure 2 depicts the system architecture including the major data flows between key components. These data flows include:

• IMU sensors in each sender collect sensory data. Each sender detects the objects captured by the on-board cameras and corresponding information such as distance, relative velocity and moving direction of the objects.
• The sender system transforms the object data from relative values to absolute values based on IMU data. For instance, the system transforms the relative velocity of a detected object to absolute velocity by adding its own velocity.
• The data are then encapsulated into VDUs (see Sec. III-B).
• The sender system encapsulates networking layer information into VDUs according to the CMR (Sec. III-C).
• The senders and receivers exchange data packets via wireless communication interfaces.
• Each receiver decides whether to forward the received packet based on a network layer filter (see Section III-C).
• Each receiver transforms the absolute values of the objects to relative values based on their own IMU data.
• Each receiver filters the received VDU based on a filtering algorithm (see Algorithm 1).
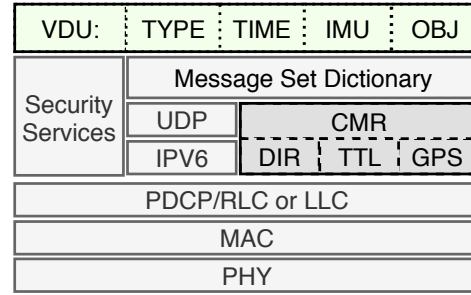
• Each receiver ARHUD renders the filtered information through AR to enhance the driver's situational awareness. The MCU perform maneuvers based on the filtered information.

As such, each vehicle is able to display the most important situational information in real-time to facilitate safe driving. Next, we describe the VDU and CMR in detail.

B. Vehicular Data Unit (VDU)

Figure 3 depicts the overall protocol stack deployed by AICP. AICP is network-agnostic and can be deployed on top of any V2V broadcast-style protocol such as DSRC. CMR is deployed as the routing protocol to provide context-aware routing in a broadcast network environment, in parallel to the traditional UDP/IP stack. Finally, the VDU contains the information required at the application layer. To accelerate data processing during filtering, we propose the vehicular data unit (VDU), a language- and platform-neutral data structure for vehicular information encapsulation. A VDU is comprised of multiple metadata fields, each of which is a key-value pair or key-value map. The metadata fields include:

• *TYPE* – whether the message is a safety or non-safety application message (pair).
• *TIME* – the time the information was first created (pair)[1].
• *IMU* – the information of IMU sensors (map), i.e., {*GPS coordinates*, *Velocity*, *diRection*, *Category*}.
• *OBJ* – the information of detected objects (map).

Similar to the SAE J2735 standard [21], the protocol stack defines a message set dictionary to specify the VDU structure and provides sufficient background information to allow vehicle systems to properly interpret the message. Together with the segmented data blocks, the dictionary extends the system compatibility by allowing different VDU structures. The dictionary and VDU also speed up the look-up process of fields such as *TIME* which can help to determine whether the received information is outdated (see Section IV).

C. Contextual Multihop Routing (CMR)

Vehicle-to-Vehicle (V2V) communication suffers from a short communication range due to signal attenuation caused by obstacles such as nearby vehicles and buildings [22]. As such, we assume that only entities within the line-of-sight

---

[1]We employ elapsed time since Unix epoch to record timestamps.

of the transmitter can receive the transmissions. Therefore, packet forwarding is crucial to extend the range over which information can be propagated. To date, most DSRC and C-V2X standards assume data transmission over a broadcast mechanism. Broadcast transmission allows a vehicle to efficiently forward information to other vehicles in their immediate vicinity. However, broadcast transmission suffers from multiple drawbacks in multihop communication. Unregulated broadcast transmission results in significant data redundancy that affects the system at every level, from increased load and congestion on the transmission medium to large amounts of unnecessary information being forwarded to the drivers. To address these concerns, we introduce CMR as the routing protocol for packet forwarding.

CMR enables the following features. (i) *Directional routing in a multihop broadcast transmission.* We consider that an object detected by a vehicle is relevant for all immediate neighbors (accessible within a single hop). However, the information is only relevant to vehicles further away from the source vehicle if the vehicles are going in the same general direction as the source. As such, vehicles only forward packets to other vehicles where this is the case. (ii) *Hop limit for geographic relevance of information.* After a certain geographic distance, the information also loses its relevance. As the number of hops acts as a rough proxy for geographic distance, a hop limit thus prevents further propagation of irrelevant information. (iii) *Lightweight message filtering.* We design the routing mechanisms to rely on only a few atomic operations in order to minimize the packet forwarding time. To provide these functions, CMR in this work relies on two different metrics, namely *GPS* and *TTL*, respectively referring to the coordinates of the detected object and a counter used to enforce the hop limit (see Figure 3). The GPS coordinates of the detected object are encoded over two fields of 32 bits to achieve precision of at least a meter.

Directional routing. We embed the heading direction and the location of the source vehicle in the packets (field *DIR* and *GPS*, respectively) to provide directional routing. The on-board magnetometer returns a 32-bit value between 0 and 360°, similar to a compass. Each vehicle receiving the message computes its own direction and compares it to the *DIR* and *GPS* fields of the message. If the receiver vehicle is heading towards the source vehicle or if both vehicles are heading in the same direction, then the packet is forwarded; otherwise, the packet is dropped. We also propose a directional antenna pattern that helps support this directional routing approach (refer to Section VI for more details).

Hop limit. Upon reception, each receiver decreases the value of *TTL* by 1. When the *TTL* reaches 0, the packet is dropped. The hop limit is defined by the original transmitter and may depend on many factors, including vehicle density, road configuration (type or geography), and the vehicle's speed. For instance, the hop limit may be higher on a crowded urban intersection with a large number of vehicles and pedestrians spread in multiple directions than on a highway during off-

TABLE I: Summary of used notations.

Symbol	Definition
$\mathcal{N}$	Set of all vehicles considered in the system
$\mathcal{V}^t$	Set of velocities $\mathcal{V}^t = \{v_1^t, \ldots, v_N^t\}$ of $\mathcal{N}$ vehicles
TTL_{th}	Initial TTL (time-to-live) of detected objects in the system
ϑ^O	Informativeness of object O
t_c	Time at which a message is created
$\mathcal{I}^i(t)$	Informativeness of message i at time t
r	Decay rate of informativeness $\mathcal{I}$ over time
$x^{j,i,o}(t)$	Binary variable indicating if object o contained in message i is received by vehicle j at time t

peak hours. In the rest of this work, we consider the hop limit to be fixed for the sake of simplicity.

Lightweight message filtering. CMR provides an initial filtering at the routing stage. However, as a routing protocol, priority should also be put on simplicity for performance reasons. As such, we design the protocol to include only a limited number of atomic operations, allowing for very fast forwarding decisions. According to the policy of CMR, each receiver drops a received message if the *DIR* difference is bigger than the threshold or the value of *TTL* is 0. Otherwise, the receiver passes the packet to the upper layers for further decapsulation while forwarding the packet via CMR broadcast. As such, CMR improves transmission efficiency by not forwarding likely irrelevant messages to the receiving vehicles.

Note that unlike common routing algorithms such as DV-CAST [23], Greedy Perimeter Stateless Routing (GPSR) [24] and its variants such as [25], [26], CMR focuses on filtering low-informativeness packets instead of improving the communication efficiency. Therefore it is a complementary protocol to the efficiency-focused routing protocols, instead of a replacement.

IV. SYSTEM MODEL AND PROBLEM FORMULATION

This section details the system model and the problem of maximizing the informativeness of the displayed objects.

A. System Model

The system includes the set $\mathcal{N} = \{1, 2, ..., N\}$ of N vehicles driving in the considered area, with velocities $\mathcal{V}^t = \{v_1^t, v_2^t, ..., v_N^t\}$ at time t, respectively. The mobility of the vehicles is exogenous to the system. Each vehicle is equipped with an ADAS or autonomous system consisting of several cameras facing varying directions for comprehensive vision around the vehicle (radar, lidar and ultrasonic sensors are optional and not a mandatory requirement of AICP), GNSS/IMU for real-time kinematic and positioning, and wireless interfaces (DSRC or C-V2X) for communications with other devices on the road. Messages are sent with a frequency between 1 and 10 Hz and the message size is limited to 300 Bytes, as specified in the C-V2X standard [13], [27]. The encapsulation and decapsulation of the messages follow the protocol standard defined in Figure 3. We model the data propagation from three parallel levels, namely the *object level*, *message level*, and *vehicle level*. *AICP* decides whether to display an object

based on the *object level* informativeness, to forward a packet based on *message level* informativeness, and targets optimizing performance based on *vehicle level* informativeness.

Object level. The processing result of each image frame contains a list of detected objects, each of which is defined as $O \triangleq \{D, V, R, C\}$, where D, V, R, C denote the *Distance, relative Velocity, diRection and Category* of the object, respectively. The rationale of the choice of these parameters is justified by the fact that an object O has a higher chance of causing an accident if (i) it is close to the vehicle, (ii) is getting closer to the vehicle, e.g., catching up with the vehicle from behind or coming right at the vehicle, and (iii) is on the heading direction of the vehicle. Additionally, the rationale for having an object category relies on the fact that certain objects could cause or sustain greater injury in an accident; e.g., a pedestrian should raise more attention than a parked vehicle or a trash bin on the side of the street.

These parameters are intertwined with each other, and such dynamics are crucial to determine a model upon which we define the informativeness of an object. For instance, mutual time and space relationship is, in fact, at the basis of modern methodologies to evaluate accidents by analyzing the collision area [28], [29]. An examination of reported accidents involving autonomous vehicles in California showed that most accidents occur at cross sections in suburban roads, with most accidents reporting rear or front damage [30]. These findings indicate that the direction with which vehicles move greatly affects the probability of an accident, especially if the vehicles are at a short distance from each other. Furthermore, we need to factor in a higher informativeness for objects that fall into the *people* category, for instance.

Upon such considerations, we express the informativeness ϑ of an object O as:

$$\vartheta^O = \left((\alpha D + \beta V)\, \gamma R \right)^{\omega C} \quad (1)$$

where α, β, γ, and ω are weighting parameters of the four priority attributes. **Note** that to improve the comprehension of the informativeness, we present a *weighted fitness sorting* algorithm (Section V-B) to calculate the inter-weight relationships using Mahalanobis distance matrix [31]. Learned from labelled datasets, the matrix realizes informativeness categorization with negligible delay.

More complicated computations such as machine learning along with accident reports of autonomous vehicles[2] [32] can incorporate the understanding of a chain of scenarios; however, they might suffer from additional costs. We leave the investigation of such an approach as possible future work.

Message level. The messages received by each vehicle may arrive at varying times and with varying delays. In fact, due to the highly time-sensitive nature of the warning messages for assisted-driving, vehicles need to establish the timeliness of the message [33]. To this end, the system extracts from the VDU the time at which a given message was created (see *TIME* in Figure 3), here denoted as t_c, and uses it to evaluate the timeliness. For a message i, its informativeness[3] can be calculated as

$$\mathcal{I}^i(t) = \left(\vartheta^i \Big(\frac{TTL^i(t)}{TTL_{th}} \, (1 - r) \Big) \right)^{(t - t_c^i)} \quad (2)$$

where $\vartheta^i = \sum_{o \in O_i} \vartheta^o$ denotes the informativeness of message i and O_i denotes the detected objects contained in message i, r is the rate at which the *informativeness* of the message decays over time, and t denotes the current time. The decay rate r is strictly connected to the time limit within which such messages are considered *up-to-date* and *relevant*. In fact, r is a system parameter that can be tuned for specific conditions. For instance, a faster decaying rate is required when conditions change quickly, such as driving on the highway at high speeds. While we need to relax the decay rate r for conditions with slower speeds (e.g., city center). For instance, a decay rate of $r = 0.15$ halves the informativeness of a message in about 4 seconds. TTL_{th} characterizes the *hop limit* defined by CMR (see Section III-C). $TTL^i(t)$ expresses the remaining time-to-live, i.e., the number of hops a message can still be forwarded, of message i at time t.

Vehicle level. At the vehicle level, a vehicle can (i) evaluate a received message as informative and display some objects from that message to the driver and subsequently broadcast the message, (ii) evaluate a received message as irrelevant to themselves but still broadcast the message, or (iii) drop the received message if its information is outdated, and thus not relevant to any vehicle in the network.

Upon receiving a multitude of messages, a vehicle derives the informativeness of the objects within. We assume a vehicle $n \in \mathcal{N}$ receives M messages at time t from other vehicles in the network, and express their informativeness as below.

$$\mathcal{I}_n(t) = \sum_{\substack{j=1 \\ j \neq n}}^{|\mathcal{N}|} \sum_{i=1}^{\mathsf{M}} \sum_{o \in O_i} \left(\vartheta^{j,i,o} \Big(\frac{TTL^{i,o}(t)}{TTL_{th}} \, (1 - r) \Big) \right)^{(t - t_c^{i,o})} x^{j,i,o}$$

$$(3)$$

The binary variable $x^{j,i,o}$ is 1 if object $o \in O_i$ contained in message i is received by vehicle j ($x^{j,i,o} = 1$), and 0 otherwise.

The number of received messages (and objects) can increase drastically when a vehicle drives into dense traffic. Displaying a large number of objects to the driver is unwise as this overloads their vision. In fact, a vehicle must first filter the incoming messages and select only those with the highest informativeness to mitigate such an issue. To this end, we next present a solution based on optimization of informativeness.

---

[3]The timeliness with which we receive given messages as well as the data contained within is strictly related to the capability to identify potential harm-causing objects. We incorporate these notions into the term *informativeness* and use it throughout the rest of the article for the sake of conciseness.

B. Problem Formulation

The following formulates an optimization problem for a vehicle to select objects whose informativeness helps to identify imminent risks, and thus increase road safety.

Specifically, the problem is defined as below.

$$\max_{j,i,o} \; \mathcal{I}_n(t) \tag{4}$$

Subject to:

$$\sum_{\substack{j=1 \\ j \neq n}}^{|\mathcal{N}|} \sum_{i=1}^{\mathsf{M}} \sum_{o \in O_i} x^{j,i,o} \leq L, \quad \forall n \in \mathcal{N} \tag{5}$$

$$TTL^{i,o}(t) > 0, \qquad\qquad \forall i \tag{6}$$

$$x^{i,j,o} \in \{0,1\}, \tag{7}$$

$$\forall j \in [1,2,...,|\mathcal{N}|], \forall i \in [1,2,...,\mathsf{M}], o \in O_i$$

The objective function in Eq. (4), defined as the summation of the informativeness of the objects contained in the messages incoming from other vehicles, expresses the *collective informativeness* that vehicle n conveys at time t. Eq. (5) limits the number of selected objects used to convey information to the driver. This allows us to display only clear and limited audio and visual content to a driver [34], [35]. Eq. (6) specifies that the time-to-live of an object must be, trivially, larger than 0. Finally, Eq. (7) restricts the $x^{j,i,o}$ variable to binary integer values. Given a suitable value of L, AICP selects L objects with the largest informativeness to display to the driver.

We also note that we assume that object resolution (also known as entity resolution) is performed before this informativeness selection step. Object resolution is the process of recognizing that multiple perceived or received objects are actually the same object just seen by multiple vehicles from different angles. As this resolution process is a well-known topic in itself we do not focus further on the process but instead refer to [36] which summarizes different object resolution methods. These methods typically have super-linear but sub-quadratic time complexity with the number of objects and thus should not significantly impact the system performance in practice. We look to study resolution techniques empirically in future work. Next we present the details of the fitness sorting algorithm in Section V, which allows such a selection with a time complexity of $O(\mathsf{M})$.

V. Prioritized Sorting Algorithm

In this section we propose the details of the filtering and prioritized sorting algorithm at the application layer. The algorithm finds the relationships between the attributes that define the informativeness to provide a more comprehensive understanding than the linear weights in Eq. (1).

A. Warm-up Radix Sorting

We present a warm up solution that orders the list $\mathbf{O^v}$ of objects recently received by a vehicle v. The radix sorting algorithm, represented by $Sort()$, arranges the orders of the tuples in $\mathbf{O^v}$. Without loss of generality, the algorithm $Sort(a)$ orders the tuples in $\mathbf{O^v}$ in ascending order in the attribute a. First, we assume the order *Distance (D)* $>_{\mathrm{I}}$ *relative Velocity (V)* $>_{\mathrm{I}}$ *diRection (R)* $>_{\mathrm{I}}$ *Category (C)* of the four attributes in a tuple. Second, we sort the tuples in $\mathbf{O^v}$ by running $Sort(C)$, $Sort(R)$, $Sort(V)$, and $Sort(D)$ sequentially. That is, we start sorting from the least significant attribute and move up to the most significant one. We observe that the running time of this solution is about $O(4n)$ since the running time of the radix sorting algorithm $Sort()$ is $O(n)$ and we need to process four attributes. However, a drawback to this solution is the assumption of a monotonic relationship between the four attributes, i.e., attribute a is more important than another b, regardless of the actual value of b.

B. Weighted Fitness Sorting

In the following we introduce a more advanced method, which not only considers the impact of the values of the four attributes, but also has a faster running time compared to the radix algorithm. To do so we first must assume that we have a labeled dataset $\mathbb{D}$ in which a large number of tuples $P_i^\star$, $i = 1, 2, ..., N$ are collected from vehicles and then labeled as either Requires Attention or Does Not Require Attention by human experts on traffic safety analysis. If $L_i^\star$ denotes the label of tuple $P_i^\star$, the labeled dataset $\mathbb{D}$ can then be represented as $\mathbb{D} = \{(P_i^\star, L_i^\star), i = 1, 2, .., N\}$. For ease of illustration, we assume $L_i^\star \in \{0, 1\}$, where 0 represents Does Not Require Attention and 1 represents Requires Attention. We note that the dataset $\mathbb{D}$ could be obtained by using the large volumes of traffic data that connected cars stream back to network centers for data analysis [37], [38]. Furthermore, there are numerous semi-supervised classification algorithms [39], [40] that can be used to build a large dataset $\mathbb{D}$ from a small initial labeled dataset which can be generated manually by traffic experts or even automakers.

Next, we discuss the details of the weighted sorting algorithm. Without loss of generality, we assume that the four attributes D, V, R and C are represented by 1-th, 2-th, 3-th and 4-th attributes, respectively. Given a tuple $P = (D, V, R, C)$ represented by $P = (x_1, x_2, x_3, x_4)$, then a labeled tuple in dataset $\mathbb{D}$ is represented by $(x_1^\star, x_2^\star, x_3^\star, x_4^\star, L_i^\star)$. To better define the relationship between the weights in Eq. (1), we define a filter $\mathbb{F}$ which weights the four attributes and their relationship by a 4×4 matrix M shown as follows

$$M = \begin{bmatrix} m_{1,1} & m_{1,2} & m_{1,3} & m_{1,4} \\ m_{2,1} & m_{2,2} & m_{2,3} & m_{2,4} \\ m_{3,1} & m_{3,2} & m_{3,3} & m_{3,4} \\ m_{4,1} & m_{4,2} & m_{4,3} & m_{4,4} \end{bmatrix}, \tag{8}$$

where $m_{i,j} > 0$ $(i, j \in \{1, 2, 3, 4\})$ is the weight assigned to the relation between the i-th and j-th attributes. Given a tuple $P = (x_1, x_2, x_3, x_4)$, filter $\mathbb{F}$ shall compute a fitness (Mahalanobis distance) as follows:

$$\mathbb{F}(P) = P \times M \times P' \tag{9}$$

Algorithm 1: AICP full-stack filtering algorithm

Networking layer filtering
thread *CMR*:

1   Determine drop or forward&process *PKT*
     according to the CMR protocol (see Sec. III-C);

Application layer filtering
thread *Fitness calculation*:

2   Update the fitness matrix M (see Eq. (8)) with
     new datasets according to Eq. (11), offline;

3 **thread** *Weighted sorting*:

4   Calculate the fitness values of the objects in
     received messages according to Eq. (9);

5   Sort the fitness values;

6 **thread** *Display*:

7   Display the first L objects in the sorted queue to
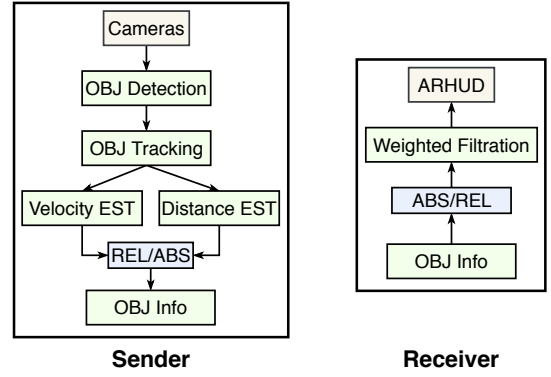   achieve Eq. (4) under the constraint of Eq. (5);

Fig. 4: Data flow of the POC prototype system.

cost is $O(1)$ to compute the fitness value of a tuple, and thus $O(N)$ to calculate the fitness values of the N tuples in the $\mathbf{O^v}$ list. In addition, the cost is $O(N)$ to sort the fitness values using the radix sorting algorithm. Hence, the total computational cost is $O(2N)$, which is less than that of the warm-up solution (which is $O(4N)$). Furthermore, the weighted solution is advantageous as it takes into account the values in each attribute and their relations. As such, we summarize the full-stack filtering algorithm of AICP in Algorithm 1.

VI. EVALUATION

A. Implementation

Following the system design in Section III, we implement a proof-of-concept (POC) prototype of AICP as a sender/receiver system. The sender detects objects in the line-of-sight and shares them with nearby vehicles in real-time. The receiver filters the received objects through the sorting algorithm presented in Section V-A and renders them to the driver in ARHUD. As shown in Figure 4, the sender consists of five components: (i) object detection, (ii) object tracking, (iii) distance estimation, (iv) velocity estimation, and (v) relative/absolute value transform; the receiver consists of (i) value transform, (ii) weighted filtration, and (iii) AR display.

We use a video recorded by the front cameras of two vehicles driven across a European capital city center, one following the other. The sender streams the video into Yolo-v5 Object Detector [20] and conducts object detection in real-time. The detection outputs tuples that consist of the positions and the labels of the objects, as well as confidence scores. Next, the system feeds the results to the *object tracking* component, which maintains a list of the objects tracked in the previous frames. First, such a component calculates the Intersection over Union (IoU), defined as $\frac{\text{area of overlap}}{\text{area of union}}$, between the objects in the previous and the latest frames. Next, it uses a greedy algorithm (quicksort) to sort the similarity according to the IoU scores [42].

The system passes the tracking result to the *velocity estimation* and *distance estimation* components simultaneously. The velocity estimation uses the historical positions of an object to estimate the speed and direction of its movement.

$$= [x_1\ x_2\ x_3\ x_4] \begin{bmatrix} m_{1,1} & m_{1,2} & m_{1,3} & m_{1,4} \\ m_{2,1} & m_{2,2} & m_{2,3} & m_{2,4} \\ m_{3,1} & m_{3,2} & m_{3,3} & m_{3,4} \\ m_{4,1} & m_{4,2} & m_{4,3} & m_{4,4} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

For ease of illustration, we use $d(P_i, P_j)$ to represent the difference of the fitness between two tuples P_i and P_j, i.e.,

$$d(P_i, P_j) = (P_i - P_j) \times M \times (P_i - P_j)' \quad (10)$$

We find suitable values for the matrix M used by filter $\mathbb{F}$ by leveraging the knowledge from the labeled dataset $\mathbb{D}$. Let $\mathbb{D}_0$ represent a subset of $\mathbb{D}$ where each tuple is labeled with 0, and $\mathbb{D}_1 = \mathbb{D} - \mathbb{D}_0$. The idea is to find a matrix $M^\star$ that maximizes the difference of the fitness between the subset $\mathbb{D}_0$ and $\mathbb{D}_1$, i.e., the fitness of the tuples in $\mathbb{D}_0$ shall be as small as possible, while that of the tuples in $\mathbb{D}_1$ can be as large as possible or vice-versa. More formally, we solve the following optimization problem.

$$\arg\max_{M} \sum_{i \neq j, L_i^\star \neq L_j^\star} d(P_i^\star, P_j^\star) - \sum_{i \neq j, L_i^\star = L_j^\star} d(P_i^\star, P_j^\star) \quad (11)$$

The formula on the left side of the subtraction sign defines the distance between two tuples with different labels (one tuple is labeled 0 and the other 1), while the formula on the right side defines the distance between two tuples with the same label (the tuples are labeled either 0 or 1). This is a classic metric learning problem for which numerous numerical optimization algorithms are available [41]. This analysis can be carried out offline, and the obtained M matrix can be used to quickly evaluate whether a newly received tuple P needs further processing by calculating its fitness as shown in Eq. (9).

Next, we analyze the time complexity of the weighted sorting solution. As mentioned, obtaining M offline allows us to neglect its computational cost. Upon obtaining M, the

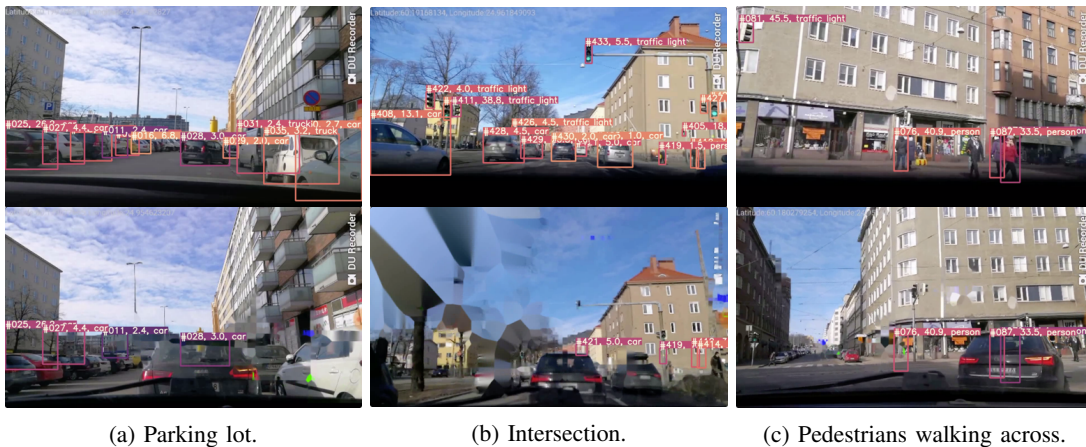(a) Parking lot.　　(b) Intersection.　　(c) Pedestrians walking across.

Fig. 5: View of the POC prototype system. At the top we see all the objects detected by a *sender* vehicle, whereas at the bottom a *receiver* vehicle sees only the objects that result from our filtering algorithm. For the sake of comprehensibility, we show all the objects that the sender vehicle detects, although a vehicle does not need to render its own objects.

The distance estimation uses the object's position and the ratio between the object area and the average size of the object to estimate the distance to the object. We categorize the distance into three classes, namely nearby, middle, and far away. After gathering the objects' positions, labels, velocities, and distances, the sender broadcasts the messages. The receiver applies the *weighted fitness sorting* algorithm to select the top L important objects out of the received message and renders them to the driver.

We calculate the Mahalanobis distance in Eq. (9) using the well-known large margin nearest neighbor (LMNN) [43] due to its simplicity and fairly well performance. Overall, the video captures 18488 images with 30 FPS (frame per second). We set the maximum number of objects to be detected as 25 objects per frame. The final number of detected objects is 75876. To prove the performance of the *weighted fitness sorting* algorithm (V-B), we first manually categorize the objects into two label groups, "require attention" and "not require attention", as described in section V-B. For simplicity, we first use a predefined policy to categorize the objects into labelled datasets. We define an object as requiring attention when it has a distance is less than 23 meters (safe breaking distance when driving at a safe speed 30mph), or velocity larger than 30, or is a pedestrian. As a result, we get 22996 and 52880 objects for the two categories, respectively. We use the `metric-learn` library contributed by the authors of [56] to implement LMNN for computing Mahalanobis distance matrix (see Eq. (8) to Eq. (10)). The matrix computation takes 225.79 seconds per 10000 objects. As described in V-B, the matrix computation is offline thus does not impact real-time system performance. Using the learned matrix, each receiver can sort 100 received objects within a millisecond on average. We open source the code of the weighted fitness sorting algorithm and the 75876-object data extracted from the video at [44].

B. Showcase Performance

Table II summarizes the latencies of the processes. As shown, the overall latency in the sender system is only 11.6

TABLE II: Latency breakdown of the POC components.

	Task	Execution Time (ms)
Sender	Pre-processing	1.1
	Object Detection	6.9
	Object Tracking	2.4
	Velocity & Distance Est.	1.2
Receiver	Sorting	1
AICP	Overall Latency	12.6

milliseconds and thus has a negligible impact on information dissemination. The additional latency added to the receiver end is mainly the sort latency, which is ~ 1 millisecond on average and does not affect the networking performance. As shown in Figure 5, AICP effectively improves the understandability of the cooperative perception system by pruning shared perception information and showing only the most critical objects. In different scenarios, e.g., in a parking lot, at an intersection, and with pedestrians crossing the street, the filtered displays are much easier to comprehend and thus provide better facilitation to driving. In comparison, cooperative perception systems without AICP would display numerous objects like in Figure 5a and Figure 5b or unimportant objects like the traffic lights in Figure 5c. Note that in different areas the system can improve performance by adapting L (the number of selected objects defined in Eq. (5)). For instance, the number can be smaller in a parking lot where less objects are moving while at an intersection it may require to display more objects. Cloud service like Google Map can be used for area identification.

VII. SIMULATION

Following the performance of the prototype shown in Section VI-B, we next test the performance of AICP through a larger scale simulation. We open source the core scripts and datasets of the simulation at [44].

TABLE III: Parameters

Simulation Parameter	Value
Radio Propagation Model	Two-Ray Interference Model [46]
Shadowing Model	Obstacle Shadowing Model [47]
IEEE 802.11p Bit Rate	6 Mbps
Transmission Power	20 mW
Noise Floor	-98 dBm
Antenna Height	1.5 m
Antenna Type	Monopole [48], Front-Rear [44]
Number of Vehicles	212 [44]
Simulation Area	4000x5000 m
Simulation Time	60 s

CMR Parameter	Value
Beacon Generation Rate	10 Hz
Hop Limit	2
Max Source Heading Direction Deviation	30°
Max Source Distance	100 m
Packet Size	102 Bytes

TABLE IV: Packet format (102 B)

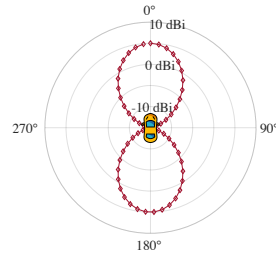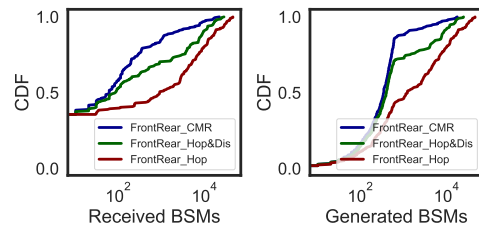| Field	Metadata	Size (Byte)
Object	ID	2
	position_x	1
	position_y	1
	velocity	1
	distance	1
	label	1
	confidence	1
Vehicle	IMUs	12
	Timestamp	2
	GPS	8

Fig. 6: Front-Rear antennae.

Fig. 7: Generated and received BSMs with different filter mechanisms and antenna modes. Hop&Dis: filtered by hops (2) and distance (100m). Hop: filtered by hops (2).

A. Simulation Setup

As pointed out in Section III-C, CMR focuses on filtering low-informativeness packets instead of communication efficiency. To isolate the effect of CMR, we exclude any communication efficiency-focused routing protocols such as GPSR [24] and DV-CAST [23]. Instead, we compare the transmission statistics with and without CMR in city-scale V2V simulations. We select an area of size 4 km by 5 km around London city center and generate traffic utilizing a real-life dataset[4].

Analogue models. We simulate the traffic during a peak period (6 pm) using `Veins` [45], an open source framework for running vehicular network simulations. `Veins` is based on `OMNeTpp`, an event-based network simulator, and `SUMO`, a road traffic simulator. We run the simulation for 60 seconds. To ensure realism we employ the two-ray interference model [46] for radio propagation. The model improves over the vanilla two-ray ground model by also capturing the ground reflection effects. We also employ the obstacle shadowing model [47] to capture the effects of buildings on signal transmissions. The upper part of Table III details the parameters.

Routing protocol. We use IEEE 802.11p as the base networking protocol for V2V communications. Following the design of CMR, we set the hop limit of each message to 2, the maximum concerned source distance to 100 meters, and the maximum heading direction difference between the source vehicle and the receiver to 30 degrees. The C-V2X standard [13], [27] recommends a message frequency between 1 and 10 Hz. To test the baseline performance, we let each vehicle broadcast VDUs at 10 Hz. From the POC test, we observe 10 objects per image on average. Thus, the average VDU packet size is set to 102 bytes. Table III and Table IV detail the CMR parameters and packet format.

B. Results

Due to the dense traffic in the selected area, the average vehicle speed in the simulation period is about 6 km/h,

[4]https://data.gov.uk/dataset/gb-road-traffic-counts

thus reflecting the lower bound of system performance under extremely congested scenarios.

Antenna type. As demonstrated by [48], [49], angled antennae, compared to idealistic isotropic antennas, can significantly change the vehicular network dynamics. Therefore, we propose a new antenna type, Front-Rear [44], to correspond with the CMR protocol. Recall that CMR prioritizes data sent by source vehicles traveling in a similar direction as the receiver. As shown in Figure 6, Front-Rear amplifies the signal in the forward and rear directions while reducing the transmission range on the sides. Hence, Front-Rear reduces the packets sent from vehicles driving on the sides and reduces the burden of the filters. Front-Rear can be deployed in a similar way as Patch [48], i.e., mounted to the front of the right and left side mirrors and the right and left side of the rear windshield.

Filters. We compare CMR with other two classical filters (hop and distance limit (Hop&Dis), and hop limit only (Hop)). Figure 7 shows the empirical cumulative distributions of the number of generated and received basic safety messages (BSMs) by vehicles when using different filters. Figure 8 and Figure 9
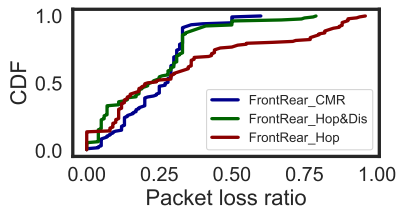
Fig. 8: Packet loss ratios with different filter mechanisms. Hop&Dis: filtered by hops (2) and distance (100m). Hop: filtered by hops (2).
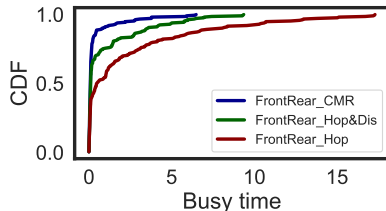

Fig. 9: Busy time with different filter mechanisms. Hop&Dis: filtered by hops (2) and distance (100m). Hop: filtered by hops (2).

TABLE V: Simulation results.

Filter	Received BSMs	Busy time (s)	Packet loss
CMR	1024	0.39	0.63
Hop&Dis	2630	1.01	0.54
Hop	5652	2.41	0.71

show the empirical cumulative distributions of the packet loss ratio and channel busy time experienced by the vehicles. As summarized in Table V, CMR effectively filters considerably more packets compared to hop&distance limit (-61%) or only hop limit (-81%). Furthermore, CMR has a slightly higher packet loss ratio than hop&distance limit, but lower than hop limit only. CMR also shows considerably less channel busy time than the other two filters (-61% and -83%, respectively).

As mentioned in Section III, AICP focuses on informativeness and thus employs CMR to filter low-informativeness packets. We implement CMR in `Veins` with only five lines of code and argue the CMR would also be lightweight in reality. Hence, *CMR could be easily integrated into routing protocols focusing on communication efficiency improvements*.

VIII. DISCUSSION

Due to the stochastic nature of human driving and the driving environment, packets containing information about a certain object may not reach their destination consistently. For instance, vehicles may move in and out of the transmission region (hop limit), and thus only receive a fraction of the packets concerning a given object. Similarly, the filtering algorithm may select different objects to display on each round. As such, the objects displayed on-screen may *flicker* and thus

significantly degrade the driving experience by distracting the driver and deteriorating the received information quality [50].

As this contradicts the goals of our proposed AICP, in future work we look to explore potential solutions such as an object persistence delay. In other words, once an object is displayed on-screen, the object remains displayed for a fixed amount of time, regardless of updates and potential filtering. This delay should be set to a value high enough in order not to distract the driver with high frequency flickering. However, longer delays may lead to a cluttering of the display with objects of low informativeness. A delay between 500 ms to 2 ms could represent an acceptable tradeoff to preserve high informativeness while avoiding flickering.

Beyond the object flickering, other user-centric and HCI aspects of the AICP system or potential system extensions could also be a target of future work. In particular, we could consider two different aspects. First, multi-modal cues (visual, audio, and tactile [35], [51]) could ease the driver's cognitive load and improve driving performance when the driver's attention primarily focuses on the road [52]. Second, we could evaluate the placement of certain visual contents (e.g., focal vs. peripheral placement [53], [54]) to determine the optimal positioning for driving performance.

Finally, our system relies on reasonable heuristics (e.g., objects physically closer to the vehicle are more important) to determine the importance of any specific object near the vehicle. However, given that a group of vehicles is an interacting set of agents, other methods might be helpful in predicting importance in more complex situations, for instance, an accident caused by a chain of actions that starts several cars away. Therefore, in future work we will examine a data-based deep learning approach that accounts for such complex situations. The motivation for this approach also derives from research showing the benefit of deep learning in related advanced driver assistance systems (ADAS) [55].

IX. CONCLUSION

In this work we propose AICP, the first solution that focuses on optimizing informativeness for pervasive cooperative perception systems with efficient filtering at both the network and application layers. To facilitate system networking, we also propose a networking protocol stack that includes VDU and CMR, a dedicated data structure and light-weight routing protocol respectively, both specifically designed for informativeness-focused applications. We also formulate the informativeness problem in cooperative perception systems from several different levels and propose a prioritized sorting algorithm for fast information-based filtering. Overall, AICP displays only the most important information shared by nearby vehicles and thus prevents information overload. We implement a POC of the proposal with ARHUD and show the system has negligible additional processing latency (12.6 ms). Additionally, simulation results show that CMR effectively filters less relevant packets, and thus considerably improves the channel availability of the vehicles.

REFERENCES

[1] THEVERGE, "California green lights fully driverless cars for testing on public roads," 2018, accessed 2018-06-15. [Online]. Available: https://www.theverge.com/2018/2/26/17054000/self-driving-car-california-dmv-regulations

[2] A. Rauch, F. Klanner, R. Rasshofer, and K. Dietmayer, "Car2x-based perception in a high-level fusion architecture for cooperative perception systems," in IEEE Intelligent Vehicles Symposium, 2012.

[3] S.-W. Kim, B. Qin, Z. J. Chong, X. Shen, W. Liu, M. H. Ang, E. Frazzoli, and D. Rus, "Multivehicle cooperative driving using cooperative perception: Design and experimental validation," IEEE Transactions on Intelligent Transportation Systems, 2014.

[4] ETSI, "Intelligent transport systems (its); Cooperative Perception Services (CPS)," ETSI TS. 103 324, Early draft, 2020.

[5] ——, "Intelligent transport system (its); Vehicular Communications; Basic Set of Applications; Analysis of the Collective-Perception Service (CPS)," ETSI TR. 103 562 v2.1.1, 2019.

[6] H. Qiu, F. Ahmad, F. Bai, M. Gruteser, and R. Govindan, "Avr: Augmented vehicular reality," in MobiSys, 2018.

[7] K. Garlichs, H.-J. Günther, and L. C. Wolf, "Generation rules for the collective perception service," in VNC, 2019.

[8] W. Liu, J. Gori, O. Rioul, M. Beaudouin-Lafon, and Y. Guiard, "How relevant is hick's law for hci?" in CHI, 2020.

[9] R. Schweickert and B. Boruff, "Short-term memory capacity: magic number or magic spell?" Journal of experimental psychology. Learning, memory, and cognition, 1986.

[10] G. Thandavarayan, M. Sepulcre, and J. Gozalvez, "Analysis of message generation rules for collective perception in connected and automated driving," in IEEE Intelligent Vehicles Symposium, 2019.

[11] S.-W. Kim, Z. J. Chong, B. Qin, X. Shen, Z. Cheng, W. Liu, and M. H. Ang, "Cooperative perception for autonomous vehicle control on the road: Motivation and experimental results," in IROS, 2013.

[12] H.-J. Günther, R. Riebl, L. Wolf, and C. Facchi, "Collective perception and decentralized congestion control in vehicular ad-hoc networks," in VNC, 2016.

[13] ETSI, "Intelligent Transport Systems (ITS); ITS-G5 Access Layer Specification for Intelligent Transport Systems Operating in the 5 GHz Frequency Band," ETSI TR. 302 663 V1.3.0, 2019.

[14] S.-W. Kim and W. Liu, "Cooperative autonomous driving: A mirror neuron inspired intention awareness and cooperative perception approach," IEEE Intelligent Transportation Systems Magazine, 2016.

[15] S. Aoki, T. Higuchi, and O. Altintas, "Cooperative perception with deep reinforcement learning for connected vehicles," arXiv preprint arXiv:2004.10927, 2020.

[16] H. Qiu, F. Ahmad, R. Govindan, M. Gruteser, F. Bai, and G. Kar, "Augmented vehicular reality: Enabling extended vision for future vehicles," in HotMobile, 2017.

[17] Q. Chen, S. Tang, Q. Yang, and S. Fu, "Cooper: Cooperative perception for connected autonomous vehicles based on 3d point clouds," in ICDCS'19.

[18] R. W. Wolcott and R. M. Eustice, "Fast lidar localization using multiresolution gaussian mixture maps," in ICRA, 2015.

[19] A. Y. Hata and D. F. Wolf, "Feature detection for vehicle localization in urban environments using a multilayer lidar," IEEE Transactions on Intelligent Transportation Systems, 2015.

[20] G. J. et al., "ultralytics/yolov5: v3.0," Aug. 2020. [Online]. Available: https://doi.org/10.5281/zenodo.3983579

[21] SAE International, "V2x communications message set dictionary," 2020.

[22] L. telcom, "In-car mobile signal attenuation measurements," LS telcom UK, Tech. Rep, 2017.

[23] O. K. Tonguz, N. Wisitpongphan, and F. Bai, "Dv-cast: A distributed vehicular broadcast protocol for vehicular ad hoc networks," IEEE Wireless Communications, 2010.

[24] B. Karp and H.-T. Kung, "Gpsr: Greedy perimeter stateless routing for wireless networks," in MobiCom, 2000.

[25] C. Lochert, M. Mauve, H. Füßler, and H. Hartenstein, "Geographic routing in city scenarios," ACM SIGMOBILE mobile computing and communications review, 2005.

[26] F. Granelli, G. Boato, D. Kliazovich, and G. Vernazza, "Enhanced gpsr routing in multi-hop vehicular communications through movement awareness," IEEE Communications Letters, 2007.

[27] GSMA, "C-V2X; Enabling intelligent transport."

[28] I. Dirnbach, T. Kubjatko, E. Kolla, J. Ondruš, and Ž. Šarić, "Methodology designed to evaluate accidents at intersection crossings with respect to forensic purposes and transport sustainability," Sustainability, 2020.

[29] T. Higuchi, M. Giordani, A. Zanella, M. Zorzi, and O. Altintas, "Value-anticipating v2v communications for cooperative perception," in 2019 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2019.

[30] F. M. Favarò, N. Nader, S. O. Eurich, M. Tripp, and N. Varadaraju, "Examining accident reports involving autonomous vehicles in california," PLoS one, 2017.

[31] R. De Maesschalck, D. Jouan-Rimbaud, and D. L. Massart, "The mahalanobis distance," Chemometrics and intelligent laboratory systems, 2000.

[32] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," in CVPR, 2020.

[33] B. Hannah Topliss, C. Harvey, and G. Burnett, "How long can a driver look? exploring time thresholds to evaluate head-up display imagery," in AutomotiveUI, 2020.

[34] F. Naujoks, Y. Forster, K. Wiedemann, and A. Neukum, "A human-machine interface for cooperative highly automated driving," in Advances in Human Aspects of Transportation, 2017.

[35] N. N. Correia and A. Tanaka, "Avui: Designing a toolkit for audiovisual interfaces," in CHI, 2017.

[36] G. Papadakis, D. Skoutas, E. Thanos, and T. Palpanas, "Blocking and filtering techniques for entity resolution: A survey," ACM Computing Surveys (CSUR), 2020.

[37] N. Cheng, F. Lyu, J. Chen, W. Xu, H. Zhou, S. Zhang, and X. S. Shen, "Big data driven vehicular networks," IEEE Network, 2018.

[38] D. J. Sun, K. Zhang, and S. Shen, "Analyzing spatiotemporal traffic line source emissions based on massive didi online car-hailing service data," Transportation Research Part D: Transport and Environment, 2018.

[39] N. F. F. D. Silva, L. F. Coletta, and E. R. Hruschka, "A survey and comparative study of tweet sentiment analysis via semi-supervised learning," ACM Computing Surveys (CSUR), 2016.

[40] I. Triguero, S. Garcia, and F. Herrera, "Self-labeled techniques for semi-supervised learning," Knowledge and Information systems, 2015.

[41] B. Kulis et al., "Metric learning: A survey," Foundations and Trends in Machine Learning, 2013.

[42] E. Bochinski, V. Eiselein, and T. Sikora, "High-speed tracking-by-detection without using image information," in AVSS, 2017.

[43] K. Q. Weinberger and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification." Journal of Machine Learning Research, 2009.

[44] P. Zhou, "AICP simulation code and datasets," 2021. [Online]. Available: https://github.com/pengyuan-zhou/AICP

[45] C. Sommer, R. German, and F. Dressler, "Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis," TMC, 2011.

[46] C. Sommer and F. Dressler, "Using the right two-ray model? a measurement based evaluation of phy models in vanets," in MobiCom, 2011.

[47] C. Sommer, D. Eckhoff, and F. Dressler, "Ivc in cities: Signal attenuation by buildings and how parked cars can improve the situation," TMC'13.

[48] D. Kornek, M. Schack, E. Slottke, O. Klemp, I. Rolfes, and T. Kürner, "Effects of antenna characteristics and placements on a vehicle-to-vehicle channel scenario," in ICC workshops, 2010.

[49] D. Eckhoff, A. Brummer, and C. Sommer, "On the impact of antenna patterns on vanet simulation," in VNC, 2016.

[50] A. Colley, J. Häkkilä, B. Pfleging, and F. Alt, "A design space for external displays on cars," in AutomotiveUI, 2017.

[51] S. Onimaru and M. Kitazaki, "Visual and tactile information to improve drivers' performance," in IEEE Virtual Reality Conference, 2010.

[52] S. Onimaru, T. Uraoka, N. Matsuzaki, and M. Kitazaki, "Cross-modal information display to improve driving performance," in VRST, 2008.

[53] R. Häuslschmid, S. Osterwald, M. Lang, and A. Butz, "Augmenting the driver's view with peripheral information on a windshield display," in IUI, 2015.

[54] I. Chaturvedi, F. H. Bijarbooneh, T. Braud, and P. Hui, "Peripheral vision: a new killer app for smart glasses," in IUI, 2019.

[55] N. Formosa, M. Quddus, S. Ison, M. Abdel-Aty, and J. Yuan, "Predicting real-time traffic conflicts using deep learning," Accident Analysis & Prevention, 2020.

[56] W. Vazelhes, C. Carey, Y. Tang, N. Vauquier, and A. Bellet, "metric-learn: Metric learning algorithms in python,"Journal of Machine Learning Research, 2020