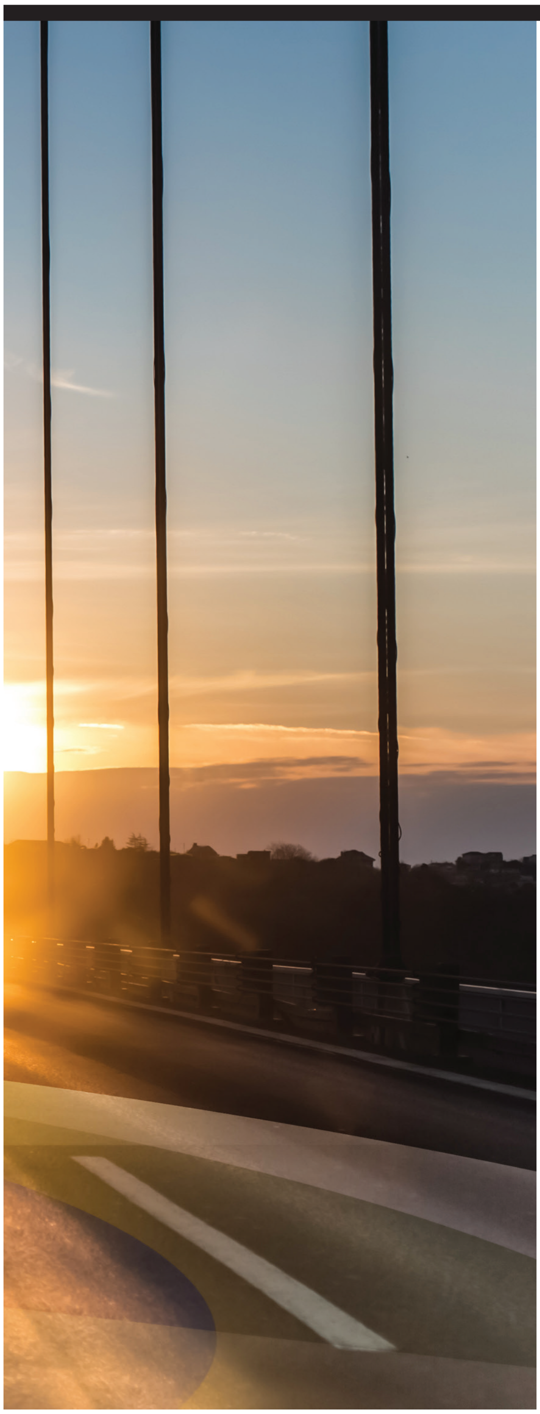
AUGMENTED VEHICULAR REALITY:

Enabling Extended Vision for Future Automobiles

Hang Qiu, Fawad Ahmad *University of Southern California*
Fan Bai *General Motors Research* **Marco Gruteser** *Rutgers University*
Ramesh Govindan *University of Southern California*



Photo, istockphoto.com

Editors: Nic Lane and Xia Zhou

Excerpted from "AVR: Augmented Vehicular Reality" from ACM MobiSys 2018, *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications and Services* with permission. https://dl.acm.org/citation.cfm?id=3210319 © ACM 2018

Autonomous vehicle prototypes today come with line-of-sight depth perception sensors like 3D cameras. These 3D sensors are used for improving vehicular safety in autonomous driving, but have fundamentally limited visibility due to occlusions, sensing range, and extreme weather and lighting conditions. To improve visibility and performance, we explore a capability called *Augmented Vehicular Reality (AVR)*. AVR broadens the vehicle's visual horizon by enabling it to wirelessly share visual information with other nearby vehicles. We show that AVR is feasible using off-the-shelf wireless technologies, and it can qualitatively change the decisions made by autonomous vehicle path planning algorithms. Our AVR prototype achieves positioning accuracies that are within a few percentages of car lengths and lane widths, and it is optimized to process frames at 30fps.

Autonomous cars are becoming a reality, but have to demonstrate reliability in the face of environmental uncertainty. Today, a few high-end vehicles have a 3D sensing capability that provides depth perception of the car's surroundings. This capability is likely to become pervasive in future vehicles and can be achieved using *3D sensors* such as advanced multi-beam LiDAR, RADAR, long-range ultrasonic and forward-facing or surrounding-view camera sensors. These 3D sensors can be used to detect and track moving objects, and to produce a high-definition (HD) map for localization [2,3]. Recent research in autonomous driving [4,5,6] has leveraged some of these advanced sensors to improve perception.

These 3D sensors have one common feature: they generate periodic 3D frames, where each frame represents the instantaneous 3D view of the environment. A 2D image frame is represented by an array of pixels, but a 3D frame is represented by a *point cloud*. Each *point* in the point cloud frame contains the *three-dimensional position* (which enables depth perception) and (optionally) the color of the point. A 64-beam Velodyne LiDAR can collect point clouds at 10 Hz containing a total of 2.2 million points each second encompassing a 360 degree view with an effective sensing range of 120 m. Stereo cameras can collect point clouds at 60 Hz with more than 55 million points per second, but with a limited field of view (110 degrees) and an effective sensing range of 20 m [7].

However, these 3D sensors only provide line-of-sight perception and obstacles can often block a vehicle's sensing range. Moreover, the effective sensing range of these sensors is often limited by weather conditions (e.g., fog, rain, snow, etc.) or lighting conditions [8, 9]. These limitations can impact the efficacy of autonomous driving or advanced driver assistance systems (ADAS). Consider the following examples. A car is following a slow-moving truck on a single-lane highway. The car would like to overtake the truck, but its 3D sensor is obstructed by the truck so it cannot see oncoming cars in the opposite lane. Similarly, two cars, waiting to turn left at an unprotected intersection, can each be "blinded" by each other. As a third example, consider a car following another one. The leader's driver may be distracted, and brake suddenly upon noticing a pedestrian entering a crosswalk. The follower, unable to see the pedestrian, cannot brake in time to prevent rear-ending the leader.

In these situations, vehicles can benefit from wirelessly sharing visual information with each other, *effectively extending their visual horizon*. This would augment vehicular visibility in hazards, and enable improved perception under challenging scenarios (Figure 1). Specifically, in the one-lane highway scenario, if the truck were to communicate visual information from its 3D sensors to the car, using some form of V2V technology, the latter's autonomous driving or ADAS software could determine the safest time and the speed at which to overtake the truck. Similarly, in the left-turn or the platoon scenarios, the transmission of visual information can help cars turn, or stop, safely.

This capability, which we call Augmented Vehicular Reality (AVR), aims to combine emerging vision technologies being developed specifically for vehicles, with off-the-shelf communication technologies. In this article, we describe the design of AVR. AVR adapts a recent feature-based SLAM technique to relatively position a receiver with respect to a sender so that the receiver can re-orient received point clouds with respect to its own position. To reduce bandwidth requirements of transferring full point clouds, AVR isolates and analyzes the motion of dynamic objects. AVR uses an adaptive transmission strategy that sends motion vectors instead of point clouds to cope with channel variability. It incorporates careful pipelining to increase the frame rate, and leverages motion prediction to hide latency.

Our initial AVR prototype can transmit visual information within 150-200 ms and process at 30 fps. It also adapts gracefully to channel variability. AVR's extended vision, when used as input to path planning algorithms, can avoid dangerous overtake attempts resulting from limited visibility.

TO IMPROVE VISIBILITY AND PERFORMANCE, WE EXPLORE A CAPABILITY CALLED AUGMENTED VEHICULAR REALITY (AVR)

AVR DESIGN

AVR consists of two logically distinct sets of components (Figure 2). One runs on a *sender* and contains the 3D frame processing algorithms that generate visual descriptions to be sent to one or more *receivers*. Receivers can either feed the received visual descriptions to a heads-up display (Figure 1c), or reconstruct an extended view containing the visual

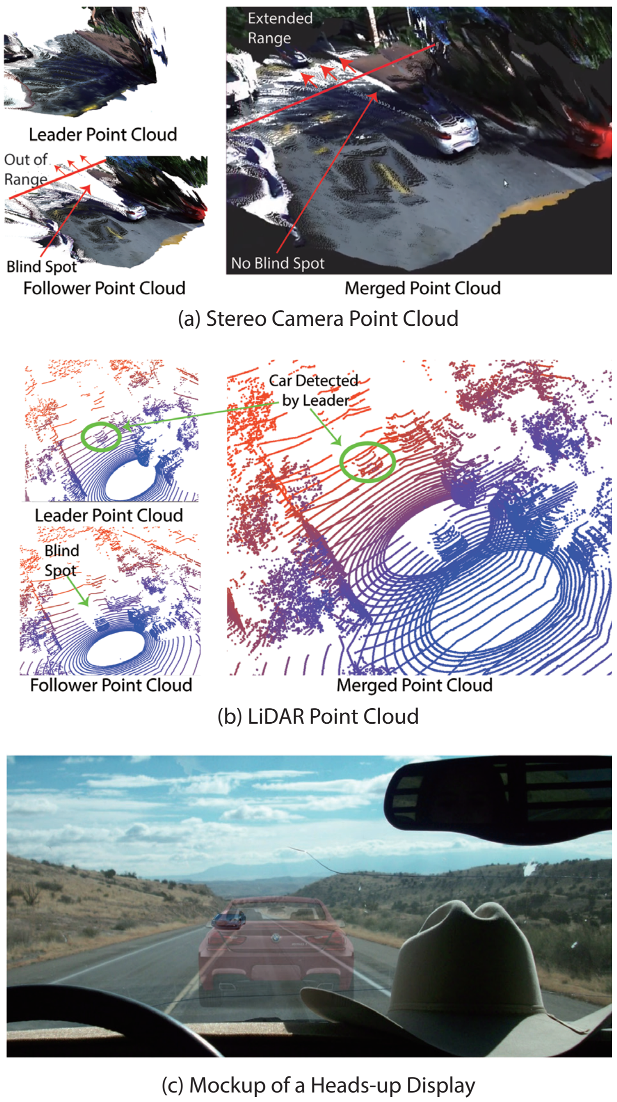


FIGURE 1. AVR allows a follower vehicle to see objects that it cannot otherwise see because it is obstructed by a leader vehicle.

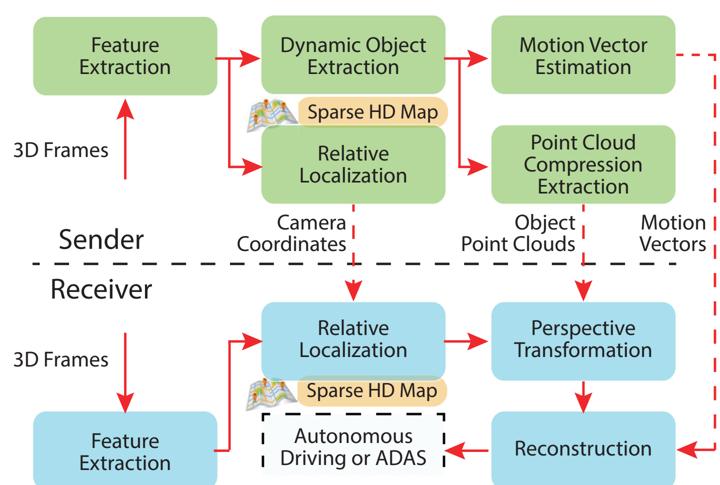
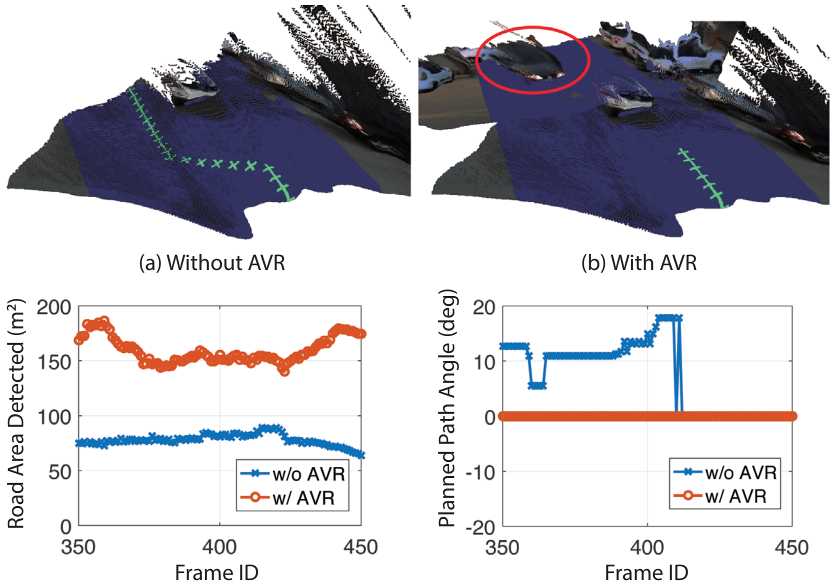


FIGURE 2. AVR sender and receiver side components.



(c) Double Road Area Detected (d) Planned Path Avoiding the Overtake

FIGURE 3. Path Planning and Road Detection in Action: a) without AVR, the follower vehicle decides to overtake the white SUV; b) with extended vision, the oncoming vehicle and more road surface area is detected, and the path planner decides not to attempt overtaking. c) AVR helps the follower detect twice the road than it might have otherwise. d) With AVR, the follower would avoid the overtake maneuver.

descriptions received from the sender with their own 3D sensor outputs. This extended view can be fed into ADAS or autonomous driving software.

AVR poses several challenges. First, for AVR, each vehicle needs to transform the received visual information into its own view. To do this, AVR must *estimate its position and orientation* with respect to the sender of the raw sensor data, and then perform a *perspective transformation* that re-orients the received point cloud.

To address this challenge, both the sender and the receiver share a *sparse 3D map* of the road and the roadside structures. This map is analogous to the 3D maps that autonomous driving systems use to position themselves with respect to the environment, but with one important difference: it is sparse, in that it only contains *features* extracted from the denser 3D maps used by these systems. For AVR, such a sparse map suffices for *relative* positioning. The sparse 3D map can be constructed offline, and potentially crowd-sourced. With this sparse map, the sender processes 3D frames from its camera and extracts features within the 3D frames, then uses some of these features to position its own camera relative to the sparse 3D map. The sender sends its position and a compressed (see below) representation of the 3D frame to the receiver. The receiver uses the sender's camera coordinates, features extracted from its own 3D sensor, and its own copy of the sparse 3D map to estimate its position relative to the sender. After decompressing the received point clouds of dynamic objects, the receiver applies a perspective transformation to these objects to position them within its own coordinate frame of reference.

Second, if AVR were to transmit 3D frames at full frame rates, the bandwidth requirement could overwhelm current and future wireless technologies. Fortunately, successive 3D frames contain significant redundancy: static objects in the environment may, in most cases, not need to be communicated between vehicles, because they may already be available in precomputed HD maps. For this reason, an AVR sender can also, instead of sending full frames, transmit point clouds representing dynamic objects (e.g., cars, pedestrians) within its field of view, and also the motion vector of these dynamic objects. AVR transmits motion vectors adaptively: when it is "falling behind" in transferring full frames, it sends motion vectors to "catch up." The receiver uses the object's motion vectors to reconstruct the object position, and superimposes the received object's point cloud onto its own 3D frame.

Third, many of the 3D sensor processing algorithms are resource-intensive, and this impacts AVR in two ways. It can limit the rate at which frames are processed (the *throughput*), and lower frame rates can impact the accuracy of algorithms that detect and track objects or that estimate position. It can also increase the latency from when a 3D frame is captured and when the corresponding point cloud is received at another vehicle. AVR selects, where possible, lightweight sensor processing algorithms, and also optimizes the processing pipelines to permit high throughput and low end-to-end latency. Its motion vectors permit receivers to *hide* latency. More optimization details of the processing pipeline can be found in our full paper [1].

AVR EVALUATION

We use a full-fledged implementation of AVR to first demonstrate the benefits of AVR and then evaluate the end-to-end performance and reconstruction accuracy. We use two laptops each with an Intel 7^{th} generation quad-core i7 CPU clocked at 4.4GHz, 16GB of DDR4 RAM and an nVidia 1080p GPU equipped with 2560 CUDA cores. We place one laptop in a leader vehicle, and the other laptop in a follower vehicle. Each laptop is connected to a ZED stereo camera and to a TP-Link Talon 7200 wireless router. The routers communicate using the wireless distribution system (WDS) mode.

The Benefits of AVR for ADAS and Autonomous Driving

Autonomous driving and ADAS systems use several building blocks including localization, object detection, drivable space detection, path planning, and so on. Many of these could benefit from AVR. We have implemented two of these algorithms, *road surface detection* and *path planning*, to demonstrate the benefits of AVR for the *overtaking* scenario, in which a follower would like to overtake a leader car, but its view is obstructed by the leader. We collected a trace with two vehicles, a leader and a follower, driving along a road, and a third oncoming vehicle in the opposite lane. We feed the reconstructed view to the receiver where both a road detection and a path planning algorithm is applied. Figure 3 (a,b) shows the detected road (marked in blue) and the planned path (marked in connected green crosses). In the first case without AVR, the follower could only see the leader's trunk and detect road surface up to the sensing range limit with occlusion. The planner found a path to overtake the leader switching to the left lane. With AVR, the follower can not only detect much more road but also the oncoming vehicle, so the path planning algorithm does not attempt the overtaking maneuver (Figure 3d). With AVR, the follower is able to detect twice as much visible road surface as without AVR (Figure 3c).

Other performance results [1] show that AVR has significant promise. It is able to transmit full point cloud frames between two cars, with each frame interspersed with a motion vector. AVR can also transmit point cloud of dynamic objects in the scene at much higher fidelity, requiring a motion vector only once in about 300 frames. The end-to-end latency is on the order of 130 ms in the latter case. Finally, AVR achieves 1.6 cm median reconstruction error for static objects and 20 cm in the 90th percentile for dynamic objects at 20 mph. More results are discussed in the full paper [1].

CONCLUSION

In this article, we have discussed the design and implementation of an AVR system, which extends vehicular vision by enabling vehicles to communicate raw 3D sensor information. AVR can be used as input to driving assistance and autonomous driving systems. The design of AVR uses a novel relative localization technique, careful perspective transformation, dynamic object isolation, latency hiding using velocity vectors and adaptive frame transmission. Our AVR prototype is flexible enough to transmit full frames or dynamic objects, achieves 200ms end-to-end delay, can reconstruct objects to within 2-10% of car lengths and lane widths, and achieve 30 fps throughput.

Currently AVR uses stereo cameras with limited range, which might affect the performance when operating in an environment with less roadside features, inclement weather, and changing lighting conditions. Fortunately, AVR does not need to match every feature in the map in order to localize a vehicle. Future work includes incorporating LiDAR device to enhance perception range and robustness, exploring lightweight or intermediate representations of the environment, and scaling AVR to a cluster of vehicles while minimizing the communication overhead. ■

Hang Qiu received his BS degree from Shanghai Jiao Tong University, Shanghai, China. He is currently a PhD candidate in electrical engineering at University of Southern California. His research interests include networked intelligence in IOT systems, mobile/edge computing, wireless networking, deep learning, and crowdsourcing systems.

Fawad Ahmad is a PhD student at the University of Southern California. He received his BSc degree from the University of Engineering and Technology, Peshawar in Pakistan. He likes building mobile systems for both human-driven and autonomous vehicles that enable more efficient and safer driving.

Fan Bai received his BS degree from Tsinghua University, Beijing, China, and MSEE and PhD degrees from University of Southern California. Currently, he is a Staff Researcher with the Electrical & Control Systems Laboratory, Research & Development and Planning, General Motors Corporation, focused on design and analysis of protocols/systems for next- generation vehicular networks, for safety, telematics, and infotainment applications.

Marco Gruteser received his MS and PhD degrees from the University of Colorado, Denver, CO. He is currently a Professor of electrical and computer engineering as well as computer science (by courtesy) at Wireless Information Network Laboratory, Rutgers University, North Brunswick, NJ. He directs research in mobile computing, pioneered location privacy techniques, and has focused on connected vehicles challenges.

Ramesh Govindan received his BTech degree from the Indian Institute of Technology Madras, Chennai, India, and his MS and PhD degrees from the University of California, Berkeley, CA. He is the Northrop Grumman Chair in Engineering and a Professor of computer science and electrical engineering with the University of Southern California, Los Angeles, CA, USA. His research interests include routing and measurements in large internets, networked sensing systems, and mobile computing systems.

REFERENCES

[1] Hang Qiu et al. "AVR: Augmented vehicular reality". In *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services*. MobiSys '18. Munich, Germany, 2018.
[2] Sebastian run et al. "Stanley: The robot that won the DARPA grand challenge". In *J. Robot. Syst.* 23.9 (Sept. 2006), pp. 661–692.
[3] Chris Urmson et al. "Autonomous driving in urban environments: Boss and the urban challenge". In *J. Field Robot.* 25.8 (Aug. 2008), pp. 425–466.
[4] Google Self-Driving Car Project Monthly Report September 2016. https://static.googleusercontent.com/media/www.google.com/en//selfdrivingcar/files/reports/report-0916.pdf.
[5] How Uber's First Self-Driving Car Works. http://www.businessinsider.com/how-ubers-driverless-cars-work-2016-9.
[6] Tesla Autopilot. http://www.businessinsider.com/how-teslas-autopilot-works-2016-7.
[7] ZED Stereo Camera. https://www.stereolabs.com/.
[8] Miguel Angel Sotelo et al. "A color vision-based lane tracking system for autonomous driving on unmarked roads." In *Auton. Robots* 16.1 (Jan. 2004), pp. 95–116.
[9] A. Gern, R. Moebus, and U. Franke. "Vision-based lane recognition under adverse weather conditions using optical flow". In *Intelligent Vehicle Symposium*, 2002. IEEE. Vol. 2. 2002, 652–657 vol.2.