Research Statement **Hang Qiu** **hangqiu@stanford.edu**

The confluence of advances in edge computing and machine learning (ML) have transformed our cyber-physical world, as we see smart edge devices, such as personal assistants, delivery drones, and autonomous cars, coming into our daily lives. Instead of using cloud services, recent years have seen these *cyber* devices embedded with remarkable machine intelligence locally, to make sense of, and sometimes react to, their *physical* world in real-time. With limited sensing and compute capabilities on-board, it is challenging yet critical to enhance the reliability of ML at the edge (Edge ML).

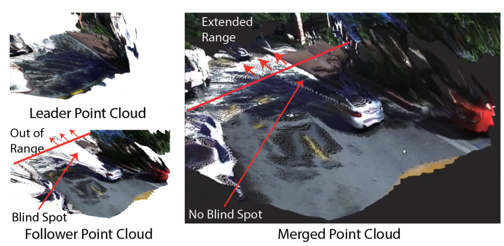
My research goal is to enable *robust collaborative intelligence* in networked cyber-physical systems. My work has identified opportunities for edge devices to *collaboratively sense, compute, and actuate*, with extra awareness of context, synthesis of information, and coordinated policies. My approach has empowered these devices to actuate with an extra layer of reliability in a novel cooperative fashion not previously demonstrated: cooperative autonomous driving vehicles with collaborative 3D perception; efficient collaborative video analytics among networked cameras for object tracking and trajectory inference; and coordinated wireless access points to avoid collision and enhance video streaming throughput. Deploying collaborative intelligence on edge devices, my recent work provides visibility into Edge ML, enabling reliable deployment, continuous monitoring, and model active learning, to further enhance the robustness of Edge ML.

More broadly, as a systems researcher, I implement my vision by building practical end-to-end systems. Solving real-world problems, I have collaborated closely with researchers in Google, Microsoft, Waymo, and GM, to contribute to industry production systems. My approach draws upon theories and methods from machine learning, wireless communication, computer vision, and robotics to answer open interdisciplinary questions at every stage in the sensing-compute-actuation loop of networked autonomous systems.

Collaborative Sensing

Autonomous agents rely heavily on sensors to perceive the environment, and yet most existing solutions focus on sensors embedded in one individual entity. As sophisticated as the sensing systems can be (such as an autonomous car with multiple LiDARs, cameras facing different directions), they are often limited by a plethora of constraints, *e.g.* line-of-sight visibility, lack or mismatch of context, lighting conditions, *etc.* To address these limitations, I developed a series of realistic collaborative sensing systems centered around autonomous cars. My recent work augments vehicle perception by sharing visual information from different perspectives between vehicles in real-time [1][6][8], and enhances localization accuracy by updating a high-definition map (HDMap) under a second [7][14]. Over a longer time-scale, my previous work also investigates vehicle sensing and localization using context-descriptive driving behaviors [4][9][10].

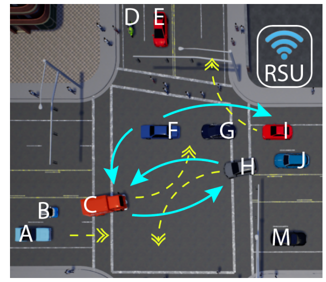
AVR: Augmented Vehicular Reality. Autonomous vehicles have *3D sensors* such as multi-beam LiDAR, RADAR, and stereo cameras. These sensors can be used to detect and track moving objects, localize and navigate autonomous cars in an instantaneous 3D view of the environment (*e.g.*, a point cloud). However, these 3D sensors only provide *line-of-sight* perception and obstacles can often block a vehicle's sensing range. To circumvent these limitations, I designed Augmented Vehicular Reality (AVR) [1], a collaborative sensing system that builds an *augmented vision* by wirelessly sharing visual information between vehicles, effectively extending the visual horizon. AVR devises novel localization techniques to position a recipient of a point cloud with respect to the sender, efficiently transform and render the point cloud in the receiver's perspective in real-time. With limited wireless capacity, AVR detects and isolates objects in the environment using a homography transformation to significantly reduce the bandwidth requirement. Further, AVR extracts motion vectors that enable reconstruction by dead-reckoning at the receiver and uses an adaptive transmission strategy that sends motion vectors instead of point clouds to cope with channel variability. AVR involves careful design of a multi-threaded pipeline that processes and transmits visual information at 30 fps with a <100ms end-to-end delay. The extended vision, when used as input to path planning algorithms, can avoid dangerous overtake attempts resulting from limited visibility.



AVR extends vision beyond line-of-sight.

This work was awarded the Mobisys best paper runner-up [1], highlighted in the GetMobile Magazine [13], and was adopted by General Motors with a global patent [19].

AutoCast: Collaborative Perception at Scale. With AutoCast [6], I have taken a major step towards scaling AVR to practical dense traffic scenarios. Take a busy intersection as an example, a left-turning car (H) needs information about cars rushing past yellow lights (A) in the opposite direction, which could be blocked by another left-turning truck (C). A right-turning car (I) needs pedestrian information that can be shared by other participants (F) or a road-side unit (RSU). Coordinating and scheduling these communications over the vehicular network before the data becomes irrelevant is challenging due to bandwidth limitations and latency constraints. AutoCast enables vehicles to efficiently and cooperatively exchange sensor data by introducing a control plane for interaction, clustering, and scheduling. In AutoCast, vehicles partition views and assess the relevance with respect to each receiver. AutoCast prioritizes important views in the schedule while eliminating duplicates and redundancies. AutoCast scales gracefully with vehicle density, enabling transmissions of 2 to 3x more useful information than sharing data in a random order.



AutoCast coordinates V2X communications in a busy intersection.

CarMap: Fast HDMap Updates. HDMap is a crucial component for fine-grained localization. Collecting this map can be tedious and expensive. The problem is even further exacerbated by seasonal changes, construction, road closures, accidents, *etc.*, which can easily make part of the map obsolete. To address this problem, CarMap [7] devises techniques for updating a feature-based HDMap in near real-time, using crowd-sourcing. In this approach, each vehicle detects map changes and uploads a compact representation to a cloud service which stitches and makes these updates available to other vehicles *within one second*. In the stitching process, CarMap robustly matches map features using both the semantics and the 3D location cues to accurately update the base map, while filtering out transient and obsolete map segments. CarMap requires 75× lower bandwidth than competing algorithms, and significantly improves the localization accuracy in cases where the map's collection and usage conditions differ dramatically.

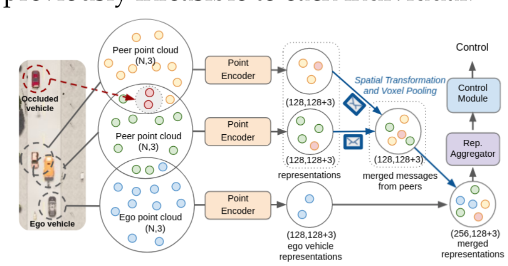
This work was also adopted by General Motors with a global patent [20].

Towards Robust Vehicular Context Sensing. At longer time-scales and distances, we developed CarLog [10] to efficiently log raw sensor data (*e.g.*, wheel angle, brake, throttle), and detect primitive driving behaviors such as hard-braking and speeding. Vehicles can compare against this historical data to detect context-descriptive driving behavior, such as changing lanes on curved roads, braking at stop signs, driving over potholes (ContextSensing [4]), as well as using their locations as anchors for localization calibration (CarLoc [9]).

Cooperative Actuation

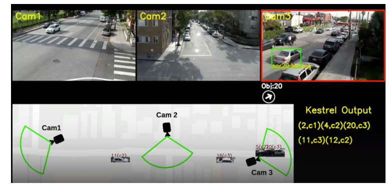
Closing the perception-control loop, my research takes the collaborative devices to the actuation stage. In addition to inventing novel ways of collaborative sensing, my research also identifies opportunities for cyber-physical agents, such as autonomous vehicles [16], surveillance cameras [3][12][18], and wireless access points [2], to cooperatively actuate, accomplishing tasks previously infeasible to each individual.

Coopernaut: Cooperative Autonomous Driving. With collaborative perception (*AVR* and *AutoCast*), the next question is how to effectively leverage shared information to make better driving decisions. The key design challenge is *what* data to transmit within the limited network bandwidth and *how* to use the aggregated information to build a coherent and accurate understanding of traffic situations. One straightforward solution is to use existing deep learning methods designed for egocentric perception but merge raw sensory data from all neighbors. This solution requires a careful selection of points to share to meet the bandwidth limit (as shown in AutoCast). Instead, Coopernaut [16] learns to encode and share meaningful scene information as compact 3D point features. The point features are physically grounded, which can be spatially transformed to heterogeneous receivers' perspectives. Using imitation learning from an A* agent with the collaborative vision from AutoCast, Coopernaut achieves a 40% improvement in average success rate over egocentric driving models and a 5x bandwidth reduction compared to the state-of-the-art cooperative perception solutions.



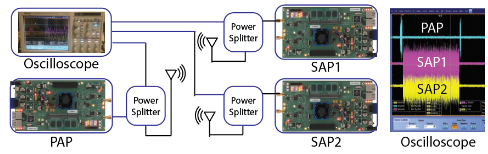
Coopernaut encodes point cloud into compact representations, broadcasts to neighbors, and fuses received messages for cooperative driving.

Kestrel: Multi-Camera Video Analytics. Cameras have become the most pervasive type of IoT sensor that generates a humongous volume of video data. The challenge in these networks is to enable *efficient video analytics*: the ability to process videos cheaply and quickly to enable searching for specific sequences of events. Kestrel [3] turns cameras into networked agents that can proactively cooperate to search and analyze videos efficiently. In particular, Kestrel tracks the path of vehicles across a heterogeneous camera network. Kestrel processes fixed camera feeds on the cloud, providing lightweight cues to mobile devices. Mobile devices are invoked only to resolve ambiguities in vehicle tracks. Kestrel squeezes deep neural networks into mobile devices to detect objects, extracts attributes using cheap visual features, resolves path ambiguities by careful association in real-time, and reduces energy budget by an order of magnitude.



Kestrel enables cameras to cooperatively track vehicles in real-time.

CoBCast: Cooperative Video Streaming. Streaming videos to mobile devices significantly stresses the current wireless network and pushes the demand for higher wireless data rates. With densely deployed access points (APs) in crowded scenarios, unicast rates remain very low due to severe interference and time-sharing. Leveraging the broadcasting nature of wireless transmissions, CoBCast [2] coordinates multiple APs to provide participants of big events with high multicast rates that can support multiple high definition video streams. CoBCast requires no complicated time or frequency synchronization and no instantaneous channel state information. Yet, despite the time and frequency offsets among concurrent transmitters, the aggregate signal from the coordinated APs offers uniform coverage and high SINR to all users. Built on software-defined radios (SDR), CoBCast revisits the underlying theory of such concurrent transmissions, and introduces several PHY techniques and a novel coordination scheme on top of a WiFi reference design. Both wireless experiments and large-scale simulations demonstrate that CoBCast can achieve 100Mbps multicast rates, orders of magnitude higher than uncoordinated transmissions.

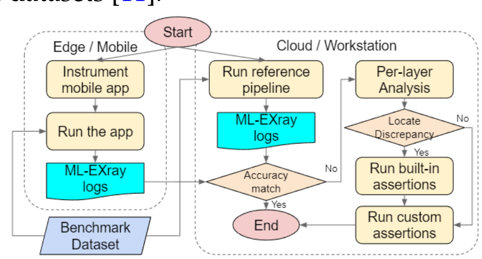


CoBCast coordinates access points to broadcast concurrently, offering uniform coverage and higher SINR to all users.

Robust Edge ML

The core component in a cooperative cyber-physical system is the Edge ML models. These models reside everywhere throughout the perception-control pipeline. Recent years have seen a shift of these models from serving on the cloud to being deployed on the actual edge devices, to enable low-latency, low-power, privacy-sensitive applications (*e.g.,* autonomous cars, personal assistants, video analytics). More often than not, however, the real-world performance can be below expectation. What exacerbates the problem is that it is very challenging to reason about the degraded performance of the black box. To improve the robustness of Edge ML, and bridge the disconnect between machine learning and system research, my research looks at how to systematically validate and debug ML deployment at the edge [5], how to automate data collection and annotation [15], learning from those unseen or out-of-distribution data in real-world in a federated manner [17], and how to minimize the cost of building such huge datasets [11].

ML-EXray: Visibility into Edge ML. When deploying ML models on actual edge devices, application developers often experience a hard time towards success, due to hardware heterogeneity, and environmental and sensory variations, *etc.* Also, design choices made by model engineers during the training process are lost in the handoff, leaving the app developers to debug with trial and error. The key challenge is that there is little visibility into ML inference execution on edge devices, and very little awareness of potential issues during the edge deployment process. ML-EXray [5] is an end-to-end validation framework that provides visibility into layer-level details of the ML execution, and helps developers debug cloud-to-edge deployment issues. Working with industry partners on production edge ML pipelines, we found that the reason for sub-optimal performance does not only lie in the model itself, but every operation throughout the data flow and the deployment process. ML-EXray effectively

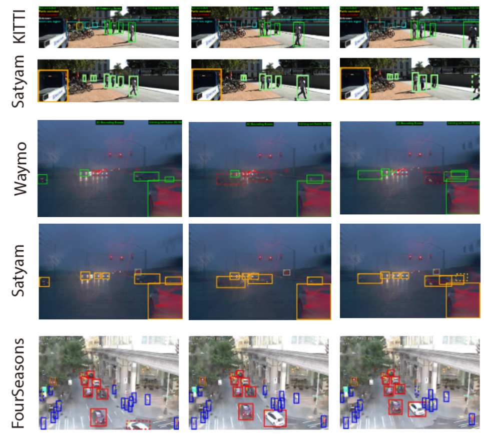


ML-EXray examines edge ML deployment, provides layer-level visibility into ML execution, and allows custom assertions for debugging.

diagnoses deployment issues, such as pre-processing bugs, quantization issues, suboptimal kernels, *etc.* Using ML-EXray, users need to write less than 15 lines of code to fully examine the edge deployment pipeline. Eradicating these issues, ML-EXray can correct model performance by up to 30%, pinpoint error-prone layers, and guide users to optimize kernel execution latency by two orders of magnitude.

This work was in collaboration with Google, and used to find deployment issues in production TFLite application. ML-EXray is released as a multi-lingual instrumentation library and an pip package.

Satyam: Automated Groundtruth at Scale. Groundtruth is crucial for testing and training machine learning (ML) based systems. Crowdtasking platforms, such as Amazon Mechanical Turk (AMT), can be used to acquire this groundtruth by obtaining results from multiple workers, and fusing these results. Automating this fusion for various ML tasks is important to reduce the burden on the researcher, but is complicated by the need to ensure high quality in the face of untrained workers, human errors, and spammers. To abstract this laborious process, Satyam [15] introduces a unified framework for *automated quality control* for complex vision tasks such as object detection, tracking, and segmentation, *etc.* Satyam also provides customizable UI templates for popular vision tasks, automates AMT task management (launching, payments, pricing, *etc.*), and filters inefficient workers. We validate Satyam's quality control techniques using several popular benchmarks (KITTI, Waymo, PascalVOC, *etc.*). Satyam achieves over 98% precision and recall and discovers up to 3% extra missing labels. Satyam has now been open-sourced for researchers to build their application-specific datasets. We have used Satyam to collect and build a trans-seasonal detection and tracking benchmark, FourSeasons. FourSeasons captures the long-term seasonal and diurnal variations for traffic surveillance cameras by providing annotations of videos spanning over a year. Researchers can use the FourSeasons Benchmark to evaluate the robustness of Edge ML models to lighting conditions, weather, and seasonal effects.



Satyam collects high quality labels for public datasets (KITTI, Waymo) in just 2 days, with 10x lower cost than labeling services (Google, Amazon). We have released FourSeasons, the world's largest trans-seasonal detection and tracking dataset.

This work was adopted by Microsoft Azure ML and was featured in Microsoft Ignite 2019.

ActiveLabeling and continuous Edge ML monitoring. As an extension to Satyam, ActiveLabeling [11] explores cost-optimal human-machine labeling. It further cuts down the labeling cost (10x) by bringing machine learning models into the labeling loop. Using active learning, the system selects the most informative samples for human labeling to train a model to label the most confident samples in the rest of the dataset. During the active learning process, ActiveLabeling automatically predicts and balances the iterative training and human labeling cost to achieve the lowest total cost. This line of research on systems for Edge ML, including ML-EXray, Satyam, and ActiveLabeling, forms a closed-loop eco-system: it can continuously monitor deployed ML models on actual edge devices, identify failure modes and data shifts, and collect and annotate data of those failure cases, and fine-tune and update the model.

Future Directions

Technological advances have always transformed society by liberating humans from tedious work while breaking boundaries for new possibilities beyond the imagination of previous generations. In the era of IoT, I enjoy the excitement of empowering smart devices with novel functionalities and applications through collaborative intelligence that can potentially take humans to the next level of efficiency and capability. Exploring the design space of these sophisticated systems requires deep and cross-disciplinary collaborative research. My research benefited from wide collaboration with over 40 researchers from 16 institutions, with expertise in systems, ML, wireless, robotics, and computer vision. These collaborations build up and extend my background across the fields. Continuing my current ongoing work, there are a few inspirations centered around the systems I built that motivate my future directions.

Cooperative Autonomous Robots. I am fascinated by the opportunity that AVR and AutoCast open up, not only to self-driving cars, but to all kinds of autonomous robots, including aerial drones, delivery rovers, warehouse shuttles, *etc.* A recent article[1] draws a comparison to Moore's law using the metric of *mileage before disengagement* in place of the number of transistors. By exponential extrapolation, it is estimated that autonomous cars can reach human-level reliability (10^8 miles/fatalities) by 2035. If we could enable cooperative autonomous agents, they can potentially drive in a fundamentally different way than human drivers. Will that be the crucial factor that changes Moore's law "coefficient" to alter the exponentiality? Coopernaut is only an initial trial integrating shared vision into control. Future autonomous agents can negotiate about intentions, trajectory plans, delivery relays, *etc.* The space is huge. But there are many technical challenges that one must address to enable this capability. How to design such complicated messages across heterogeneous agents? How to combine received messages to feed into the control pipeline? How to deal with various delays of different copies? What kind of model should we design to process this data? To address these questions, I plan to investigate the design and implementation of the control stack by using Satyam and photo-realistic simulator (*e.g.,* Carla, AirSim) to build a *cooperative robot benchmark*. Beyond basic perception tasks like object detection, tracking, and segmentation, the benchmark encompasses human maneuver demonstrations for heterogeneous agents, provided with extended vision as well as future trajectories of other traffic participants. I have written part of the ideas along this direction into an NSF proposal, which has been granted recently[2]. To date, it has been a challenge to consistently validate autonomous driving performances. With this dataset, I can build a benchmark suite to explore and verify the tradeoff between representation sharing granularity and cooperative driving performance, design efficient neural models that can consume various asynchronous representations at different processing stages.

The Distributed Control Plane for Edge ML. In addition to autonomous robots, Edge ML is deployed to various cyber-physical devices, such as surveillance cameras, immersive AR headsets, traffic lights, cellular towers. With the promise of 5G connecting everything, there emerge collaboration opportunities among robots and these edge devices as well. To enable device collaboration, we lack the abstraction of a control plane across all edge devices, and a programming model to access other devices. Further, the mobility of agents requires the coordination of the control plane to occur dynamically in a potentially distributed fashion when no static edge resource is available. In my future work, I plan to investigate the feasibility of dynamically generated control plane for instantaneously formed clusters of edge devices, drawing theoretical analysis from clustering and scheduling algorithms, and develop realistic coordination and cooperative systems with novel applications in communication, transportation, remote operation, *etc.* This distributed control plane substrate can be developed into an open-source framework to allow researchers to connect IoT devices, develop programming languages for their communication, and innovate new applications. One ultimate vision of this distributed control plane is to enable IoT devices to talk to any relevant devices for help, enabling self-supervised federated learning that mines the errors and corner cases from continuous streams of multi-agent sensor data. Given the possibility of constant connection, agents should develop skills to ask the right question to the right device, harvest new knowledge to generate iterative queries, and seek a pathway to achieve their goals with minimum overhead.

Robustness and Security. Autonomous sensing-compute-actuation systems demand robust decision making, and my work in ML-EXray and FourSeasons demonstrates how vulnerable a vision-based deep learning model can be to abnormal changes and corner cases. Meanwhile, ML-EXray and Satyam put me in a particularly advantageous point to investigate the robustness of general machine learning models to weather, seasonal, and lighting changes. When models learn to take in collaborative sensing results, it further challenges the robustness to the variability of the exchanged information. Also, in terms of security, even though collaboration promises more informed decisions, communication creates a new attack surface. With built-in security safeguards, an adversary could still compromise and gain complete control of a set of entities, and use them to inject false data to force other agents to behave maliciously, taking unsafe or undesirable actions and tampering with other network-connected units. How can we detect malicious attacks in this case? Can inconsistency between multiple received copies help? What kind of redundant information can we add as credentials? How much overhead will that incur? How to quickly recover from the attack? Those questions have to be answered before collaborative sensing-compute-actuation systems can penetrate. Therefore, I intend to investigate how to ensure robustness and security at the sensing, compute, and actuation stage of autonomous cyber-physical systems.

[1] The Moore's Law for Self-Driving Vehicles

[2] NSF Grant: CNS Core: Medium: Network-Enabled Cooperative Perception for Future Autonomous Vehicles

Access to Research and Research Artifacts

When possible, I have released all of my publications, code, and datasets in public domain.

- AVR: https://github.com/hangqiu/AVR16
- ML-EXray: https://test.pypi.org/project/MLEXray/
- AutoCast: https://autocastnet.wordpress.com/
- CarMap: https://github.com/USC-NSL/CarMap
- Satyam: https://github.com/satyamresearch/satyam, Web: https://researchsatyam.wordpress.com
- FourSeasons Benchmark: https://trafficcamdataset.wordpress.com

My Publication (Selected)

[1] AVR: Augmented Vehicular Reality
Hang Qiu, Fawad Ahmad, Fan Bai, Marco Gruteser, Ramesh Govindan
Proceedings of the 16th ACM International Conference on Mobile Systems, Applications, and Services (MobiSys'18), Munich, Germany, June 2018.
Best Paper Runner-up Award

[2] CoBCast: High-rate WiFi Broadcasting in Crowded Scenarios via Lightweight Coordination of Multiple Access Points
Hang Qiu, Konstantinos Psounis, Giuseppe Caire, Keith M. Chugg, Kaidong Wang
Proceedings of the 17th International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'16), Paderborn, Germany, July 2016.

[3] Kestrel: Video Analytics for Augmented Multi-Camera Vehicle Tracking
Hang Qiu, Xiaochen Liu, Swati Rallapalli, Archith Bency, Rahul Urgaonkar, B.S. Manjunath, Ramesh Govindan, Kevin Chan
Proceedings of the 3rd ACM/IEEE International Conference on Internet-of-Things Design and Implementation (IoTDI'18), Orlando, Florida, April 2018.

[4] Towards Robust Vehicular Context Sensing
Hang Qiu, Jinzhu Chen, Matt McCartney, Yurong Jiang, Shubham Jain, Gorkem Kar, Donald Grimm, Fan Bai, Marco Gruteser, Ramesh Govindan
IEEE Transactions on Vehicular Technology (IEEE TVT), vol. 67, no. 3, pp. 1909-1922, March 2018.

[5] ML-EXray: Visibility into ML Deployment on the Edge
Hang Qiu, Ioanna Vavelidou, Jian Li, Evgenya Pergament, Pete Warden, Sandeep Chinchali, Zain Asgar, Sachin Katti
Proceedings of the 5th Conference on Machine Learning and Systems (MLSys'22), Jan 2022.

[6] AutoCast: Scalable and Efficient Sensor Sharing between Autonomous Vehicles
Hang Qiu, Pohan Huang, Konstantinos Psounis, Ramesh Govindan
Under submission.

[7] CarMap: Fast 3D Feature Map Updates for Automobiles
Fawad Ahmad, **Hang Qiu**, Ray Eells, Fan Bai, Ramesh Govindan
Proceedings of the 17th USENIX Symposium on Networked Systems Design & Implementation (NSDI'20), Santa Clara, California, February 2020.

[8] Augmented Vehicular Reality: Enabling Extended Vision for Future Vehicles
Hang Qiu, Fawad Ahmad, Ramesh Govindan, Marco Gruteser, Fan. Bai, Gorkem Kar
Proceedings of the 18th International Workshop on Mobile Computing Systems and Applications (HotMobile'17), Sonoma, California, February 2017.

[9] CarLoc: Precisely Tracking Automobile Position
Yurong Jiang, **Hang Qiu**, Matt McCartney, Gaurav Sukhatme, Marco Gruteser, Fan Bai, Donald Grimm, Ramesh Govindan
Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems (SenSys'15), Seoul, South Korea, November 2015.

[10] CarLog: A Platform for Flexible and Efficient Automotive Sensing
Yurong. Jiang, **Hang Qiu**, Matt McCartney, William G.J. Halfond, Fan Bai, Donald Grimm, Ramesh. Govindan
Proceedings of the 12th ACM Conference on Embedded Networked Sensor Systems (SenSys'14), Memphis, Tennessee, November 2014.

[11] Minimum Cost Active Labeling
Hang Qiu, Krishna Chintalapudi, Ramesh Govindan
Under submission.

[12] On Tracking Realistic Targets in a Megacity with Contested Domain Access
Jongdeog Lee, Tarek Abdelzaher, **Hang Qiu**, Ramesh Govindan, Kelvin Marcus, Reginald Hobbs, Niranjan Suri, Will Dron
Military Communications Conference (MILCOM'18), Los Angeles, California, October 2018.

[13] Augmented Vehicular Reality: Enabling Extended Vision for Future Automobiles

Hang Qiu, Fawad Ahmad, Fan Bai, Marco Gruteser, Ramesh Govindan
GetMobile: Mobile Computing and Communications, vol. 22, issue. 4, pp. 30-34, December 2018.
Invited article

[14] QuickSketch: Building 3D Representations in Unknown Environments using Crowdsourcing
Fawad Ahmad, **Hang Qiu**, Xiaochen Liu, Fan Bai, Ramesh Govindan
Proceedings of the 21st International Conference on Information Fusion (FUSION'18), July 2018.
Invited paper

[15] Satyam: Democratizing Groundtruth for Machine Vision
Hang Qiu, Krishna Chintalapudi, Ramesh Govindan
Integrated into Microsoft Azure ML. Featured in Microsoft Ignite 2019.
Used by researchers at UCSB, USC, UIUC, ARL across different disciplines.

[16] Coopernaut: End-to-End Driving with Cooperative Perception for Networked Vehicles
Hang Qiu*, Jiaxun Cui*, Dian Chen, Peter Stone, Yuke Zhu
Under submission. Available upon request.

[17] Fedml: A research library and benchmark for federated machine learning
Chaoyang He, Songze Li, Jinhyun So, Mi Zhang, Hongyi Wang, Xiaoyang Wang, Praneeth Vepakomma, Abhishek Singh, **Hang Qiu**, Xinghua Zhu, Jianzong Wang, Shen Li, Peilin Zhao, Yan Kang, Yang Liu, Ramesh Raskar, and others
Advances in Neural Information Processing Systems (NIPS'20), SpicyFL workshop, 2020.
Best Paper Award

[18] Optimal Resource Allocation for Crowdsourced Image Processing
Kristina Wheatman, Fidan Mehmeti, Mark Mahon, **Hang Qiu**, Kevin Chan, Thomas La Porta
Proceedings of the 17th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON'20)

[19] Method and apparatus of networked scene rendering and augmentation in vehicular environments in autonomous driving
Hang Qiu, Ramesh Govindan, Marco Gruteser, Fan Bai
Worldwide Patent: US20180261095 / CN108574929 / DE102018105293

[20] Crowd-sensed Point Cloud Map
Fawad Ahmad, **Hang Qiu**, Fan Bai, Ramesh Govindan
Worldwide Patent: US20190266748 / CN110186467 / DE102019104482