Remote Driving: A Ready-to-go Approach to Autonomous Car? - Opportunities and Challenges

Ruilin Liu†, Kostas Bekris†, Ahmed Elgammal†, Vinod Ganapathy†, Mario Gerla‡, Liviu Iftode†, Melchi Michel†, and Jingang Yi†

†Rutgers, The State University of New Jersey, ‡University of California, Los Angeles

In contrast to autonomous driving solutions purely based on artificial intelligence, this article discusses an alternative for autonomous cars, which involves remote over-the-network driving. The progress in ITS technology, especially in wireless networks and robotic control theory, allows envisioning tele-operated cars that are equally effective in navigating roads and highways to cars controlled by an on-board human driver. This article also examines the challenges that remote driving needs to overcome and possible solutions before turning this technology into reality.

Introduction

Autonomous driving is broadly recognized to be a solution to the significant costs of standard automobile technology. As of 2014, both major automotive manufacturers and IT companies have been testing their driverless car systems. Google launched their self-driving car in 2012 and, by April 2014, the platform had logged nearly 700,000 autonomous miles [1]. More recently, Google revealed a 100% autonomous prototype with no steering wheel, gas pedal, or brakes. Though not as ambitious as Google's fully autonomous car, traditional automobile makers prefer an incremental approach, where they gradually introduce autonomous or semi-autonomous solutions in the driving experience. For example, A few 2014 BMW models have featured traffic-jam assistance, which could control the speed and the steering of the car in dense traffic.

Current autonomous cars rely primarily on artificial intelligence (AI) to understand the driving environment and act as human. Despite significant progress in this direction, the wide adoption of self-driving cars is far into the future as the understanding of the driving environment and autonomous planning correspond to daunting AI tasks, where human skill is not yet attained. Complex situations, which people manage effectively, cause failures for computerized autonomous systems. The later demand a huge effort in collecting and maintaining accurate maps [2]. In addition, AI based solutions require huge equipment investments. For instance, each Google car requires $150K in external sensors [2]. Only 20% of vehicle owners are interested in purchasing a fully autonomous car if it costs more than $3K than a regular one [3].

On the other hand, teleoperation technology progresses rapidly recently. Sensing data fusion from multiple sources enables remote operators to have better environment perception [4]. The representation of the environment has improved from 2D direct video capture to 3D environment visualization [5, 6]. Moreover, the vehicle controlled nowadays has gone beyond military or dedicated ground vehicle into commodity cars [7].

Inspired by this trend, we envision a different approach to driverless cars on the road – remote driving service, i.e., human operators reliably driving vehicles on the real transportation network from remote "driving centers". This can be achieved using wireless communication, appropriate interfaces and sensing information as well as short-term autonomy. The envisioned solution involves cars with relatively inexpensive on-board cameras and sensors transmitting data to a remote driving center. There, a human driver using an interface similar to today's car-driving simulators operates the car remotely. The commands are transmitted in a secure manner back to the vehicle, which uses minimal autonomy to follow the directions and guarantee safety. If feasible, remote driving can be profoundly transformative. Transportation of goods, return of unoccupied vehicles, supervision and fail-safe operation of passenger or autonomous cars are example applications.

This article explores the fundamental aspects of designing and building remotely driven vehicles and driving centers. These include the provision of real-time, safety, reliability, security guarantees and effective human interfaces. While autonomy is justifiably attracting attention, remote operation is complementary to self-driving cars and can be used to provide human oversight. By bringing human operators with sophisticated perceptual and cognitive skills into the control loop, remote driving solves challenging aspects of autonomy. Furthermore, less expensive sensors may be sufficient for remote driving relative to complete autonomy. This allows existing cars with limited additions to be used for this purpose. At the same time, the technology provides job opportunities for professional drivers in better working environments.

There are many scientific and engineering challenges in bringing this technology into reality, which lie at the intersection of networking, control and perceptual sciences. These roadblocks will be discussed in a layered manner after providing the design of the remote driving system.

Remote driving as a service

Remote driving as a service can either offer the remote control station to a special group of users to control their own vehicles or directly provide services to deliver goods or cars to the requested destination. A person needs to pass the remote driving training courses in order to be qualified for remote driving. It may be desirable in many cases for the vehicles to be operated on dedicated low-speed lanes and to clearly indicate to physical drivers of other vehicles that they are remotely driven vehicles so as to increase the safety of their operation.

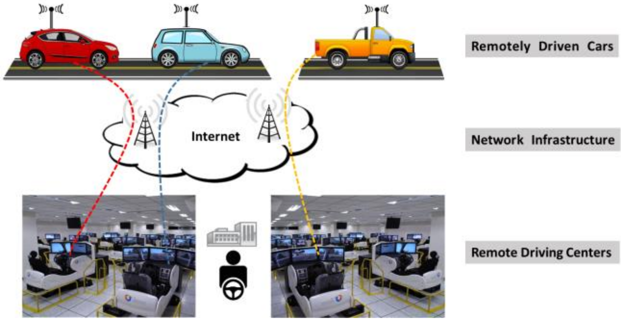

Figure 1 *Illustration of a typical remote driving system*

Remote driving has many potentially useful scenarios in our individual daily life, business-wise or for assistance purposes:

- Transporting goods. Goods transportation is boring especially for the long distance travels, where the driver needs to work in poor environment for days. Remote driving service improves the driver's working condition greatly without deteriorating their work efficiency or incurring additional costs.
- Transporting passenger cars. We envision that the future vehicles may operate in dual-mode, either physical or remote driving. Car rental business can use remote driving to deliver or get back the rented cars with less cost. Family members can also reuse the vehicle no matter where the car is parked.
- Supervised or assistant driving. A remote driver can monitor, assist capability-impeded driver, e.g. the minor or elder, or even take over the full control from the onboard driver, should the onboard driver be unable to continue driving. The stronger capability of environment perception (e.g. night vision cameras) makes supervised driving useful even for ordinary driver under severe conditions.

A remote driving system (Fig. 1) should involve three components: the remotely driven car, the network infrastructure, and the remote-driving center, where qualified drivers work and provide remote driving services.

The remotely driven car is equipped with an Onboard Proxy System (OPS), which includes multiple sensors and actuators and works as the proxy of the remote driver. The functionality of the OPS is two-fold: i) sensing the driving environment and ii) executing the driving commands. The remote-driving center contains multiple Remote Control Stations (RCS), which create the illusion that the remote driver is sitting inside the car through multi-dimensional interfaces and provision of sensing information. Moreover, the station is also responsible to translate the driver's analogue operation signals, such as steering, into digital commands that can be interpreted by the OPS. The data transmissions between the RCSs and the OPSs are achieved over the Internet.

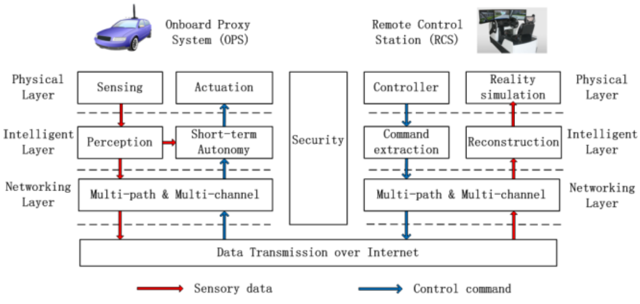

Figure 2 *Layered design for remote driving system.*

We believe that the design of the remote driving system should be designed in a layered fashion as shown in Figure 2, following the service architecture of remote driving.

The physical layer works as the interface between the remote driving system and the real world. On the OPS side, it consists of modules sensing the driving environment, and actuation modules manipulating the vehicle given operation instructions. On the RCS side, the simulator exposes the remote driver to a computer-simulated driving environment, while the controller detects and transmits driving signals into the lower layers.

The main functionality of the intelligent layer can be divided into perception functions and short-term autonomy. Furthermore, in order to reduce the bandwidth consumption, the intelligent layer is also responsible to extract the semantic information from the sensory data, use it to use it to perform data compression and then reconstruct the setup in the remote center.

The networking layer is responsible to maintain the reliable end-to-end connectivity between OPS and RCS with minimized delay. It should overcome the noise of the wireless channels as well as periodical disconnections due to frequent hand-offs in vehicular communications.

Finally, in a safety-critical system as remote driving, the security component involves multiple layers from physical hardware to data transmission over the internet.

The remainder of the article discusses the state-of-art technologies and design issues in the definition of remote driving systems. As a critical part of the remote driving system, a robust network layer is discussed first. Based on the networking layer, a proposed method to hide network delays, called "look-ahead driving", together with required sensing and autonomy technologies, are discussed next. Finally, security issues are presented before the conclusion of this article.

Robust networking

The remote driver's Quality of Experience (QoE) is primarily determined by the networking performance in transmission delay, robust connectivity and the network capacity to ensure enough end-to-end bandwidth between the RCS and OPS. The network architecture of the remote driving system consists of both wired and wireless infrastructures. The wired network generally provides adequate bandwidth and latency given a reasonable distribution of the remote driving jobs among different driving centers. The last wireless hop (or a few hops) connecting the AP (Access Point, also referred to as Base Station or Road Side Unit) to the vehicle is the most critical and unreliable segment. To solve this problem, the following technologies should be incorporated into the remote driving system.

Multi-channel transmission

We believe that redundant wireless modules should be mounted on the remotely driven car to achieve robust connectivity. WiFi and DSRC modules enable both vehicle-to-infrastructure (V2I) and vehicle-to-vehicle (V2V) communications, which help expand the coverage of sparsely distributed APs. Given the limited availability of DSRC RSUs deployed and the potential vulnerability of WiFi APs, the bulk of remote-driving traffic may also use LTE base stations in the near future. The overhead imposed on current cellular systems can be alleviated by LTE picocells to explore the spatial reuse of cellular spectrum. For example, LTE Open Base Stations are becoming popular, and are viewed as an efficient way to stretch the limited spectrum.

While LTE, WiFi and DSRC are normal options for connectivity, additional channels, such as TV white spectrum and satellite communications, may also be considered in emergencies. Although remote driving would be limited on these channels due to low bandwidth and high delay, they may be useful for the remote supervision of the vehicle's movement together with on-board autonomy.

Physical and MAC layer design for various channels should be examined to provide QoS for various message types. For example, the LTE dedicated channel guarantees bandwidth at a high cost, while the low-cost common channel causes channel quality uncertainty due to the competition among users. WiFi uses free spectrum but is subject to congestion and denial-of-service attacks. DSRC uses dedicated spectrum and favors priority messages, but data must be delivered via Road Side Units (RSUs). Thus, multi-channel transmission needs heterogeneous MAC and PHY layers, which provide the opportunity and challenge to tune configurations to improve transmission.

Reliable multi-path protocol

Providing multi-channel transmission is insufficient because of the frequent hand-offs between APs. With a single end-to-end path, a transport layer restart is needed when the channel becomes temporarily unavailable due to handoff or path blockage. The switching time from one channel to another may lead to large time gaps that undermine remote driving.

We believe that the multi-path protocols (e.g., SCTP [8] and MPTCP [9]) that maintain several active end-to-end paths simultaneously should be applied. These protocols split a single E2E session onto multiple TCP sub-flows, one per physical path, functionally independent of the main flow that feeds them. Sub-flows may be added or dropped as the user moves in or out of the coverage of an AP without disrupting the main connection.

For additional robustness, MPTCP with Network Coding (e.g., [10]) can also be applied. Network coding allows the extraction of the missing packet from other coded packets with no little delay. It can be applied at either E2E or link level. For example, the random linear network coding [10], is an E2Ed solution and it maintains the transparency at the middle points, which prevents possible pollution attacks from non-trusted APs and simplifies implementation. On the other hand, link level coding (e.g., Forward Error Correction) can be useful when a particular link of the network is vulnerable, e.g., the last hop of wireless communication in remote driving.

While MPTCP enables middle box transparency and congestion control, it has drawbacks, such as extra latency introduced by the unnecessary ARQ mechanism and redundant bandwidth consumption. UDP based multipath protocols should be considered as an alternative. For example, RTP has been proposed [11] in media sessions using single-path UDP. To implement such protocols, issues about congestion control, packet skew and path choice must be dealt with. Features, such as network coding, can be used to enhance robustness. Assuming that middle box issues affecting UDP can be avoided (for instance using TCP sublows), a bidirectional E2E UDP protocol running on multiple paths can be potentially more efficient.

Look-ahead driving

In the ideal scenario that the network provides robust connectivity with no transmission delay, remote driving is no different than on-board driving expect that it geographically separates the driver from the car. Unfortunately, there will be transmission delays experienced between the car and the remote driver.

As illustrated in Figure 3, the driving command suffers from control delays if the driver makes a decision directly using the latest sensory data transmitted from the remotely driven car. To solve this problem, we believe that a "look-ahead driving" strategy should be implemented in a remote driving system. Instead of providing to the remote driver the sensed scenes, the remote driving system should present a sequence of predicted scenes to the remote driver based on the estimated control delay δ_c and received sensory data. It essentially hides the network delay from the remote driver.

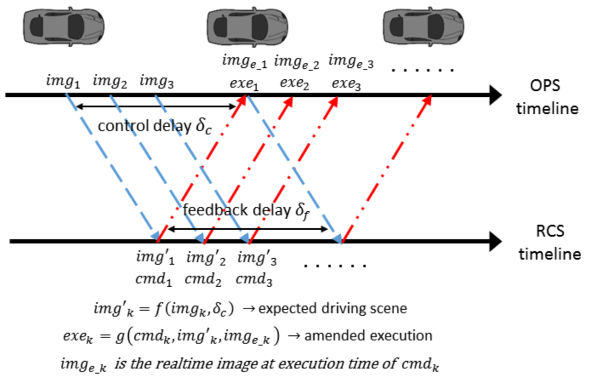

***Figure 3** The principle of "look-ahead driving"*

The "look-ahead driving" principle requires that the remote driving system should incorporate some intelligent functionalities: (a) the sensing & perception components to process the sensed data to get a predicted state of the world at the time the operator's commands will arrive on the vehicle's side and (b) the vehicle-side physical control & short-term autonomy components for determining whether the control commands generated based on predicted scenes can be trusted to ensure the vehicle's safety. Note that remote driving greatly alleviates the workload of robotic control as a remote driver provides the high level instructions, and safety (i.e., collision avoidance) is the main concern of on-board autonomy without focusing on the efficiency of motion planning (i.e., finding the best path).

Predicting dynamic scenes

If the vehicle is driving in an environment with no other cars or pedestrians, look-ahead driving reduces to estimating the location of the vehicle/camera after a delta time, depending on speed and expected delay. In a simple scenario, if the car drives in a straight trajectory, this mainly reduces to zooming-in in the field of view. To predict a dynamic scene with other vehicles, state-of-the-art technologies in segmentation, tracking and predicting moving objects (other vehicles) should be applied.

Real-time segmentation and tracking of independently moving objects can nowadays be facilitated using online algorithms [12, 13] as shown in Figure 4. These approaches formulate motion segmentation as a manifold separation problem of feature point trajectories and theoretically showed that trajectories of tracked features of the same object (rigid or non-rigid) form a non-linear manifold. On top of the segmentation and tracking, low-level Bayesian scene representation associated with a high-level semantic representation can be maintained based on motion, depth, and appearance discontinuity cues [12]. Dense scene layer segmentation is based on an optimization framework that integrates sparse motion and depth discontinuity cues with dense appearance cues [12]. The scene representation will also use classifiers to reason about scene entities at a semantic level.

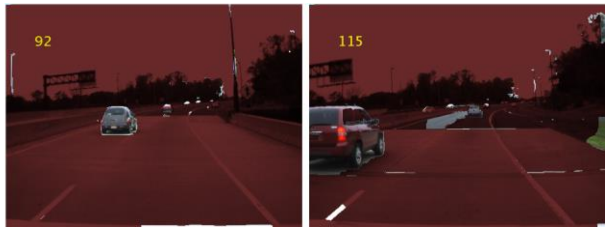

***Figure 4** Online segmentation and tracking from a vehicular camera [12]*

Short-term autonomy

Controlling dynamical systems is challenging both for automated algorithms and people. Much of the requisite skill involves an accurate and continuous perception of the system's safety limits, and how these limits project themselves onto controls. This is even harder when the operator is in a remote location and receives information over a network. The above observations make us believe that an autonomous mechanism should be incorporated in the remotely driven car to ensure safety. The autonomous mechanism can be divided into the following situations:

- If network connectivity is lost, then "emergency autonomy" is initiated, which requires safe planning under dynamics to bring the car to a complete stop until connectivity is restored.
- If connectivity exists but delays are significant, then "short-term autonomy" is required to potentially amend the human commands to the latest model of the driving environment.
- "Safety-enforcing autonomy" is continuously applied to minimize failures, such as collisions, the tires leaving the road or losing traction. All "points of no return" for the system must be avoided.

Safe planning with dynamics

Under "emergency autonomy" and for dynamical models like the above, integrated planning and control will provide the motion without human support. A planner should be called first to generate a desired trajectory given the model of the world and the closest location to park. Modern and efficient kinodynamic planners [14] should be used to provide high-quality solutions for systems with dynamics. These planners can be used in a model predictive control (MPC) framework with safety guarantees for systems with drift [15]. Then a sensing-based maneuvering system could be used to follow the trajectory with a self-pacing velocity. To design the velocity profile, a time suspension technique will be used [16] together with MPC [16, 17] to follow the desired motion profile.

Vehicle/tire dynamics and constraints are easily implemented in MPC and there are fast-computed MPC optimization solutions for real-time implementation.

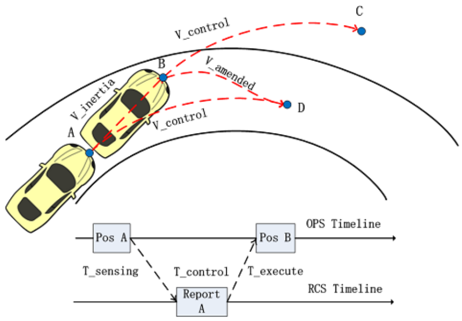

Figure 5 *Amended path planning.*

Amending delayed human controls

When human inputs are available, they have to be evaluated whether they have taken into account a considerably older version of the driving environment due to delays. Figure 5 shows an example, where a vehicle follows a curve and network delays cause the human control to be outdated by the time it arrives on the car, assuming no scene prediction is enforced.

The "look-ahead driving" principle will help reduce such situations but adaptation of the driving commands will need to take place in the general case. To achieve this objective the planning process will minimize the difference to the outcome of the human provided controls while guaranteeing the safety of the vehicle, given the latest perception of the world. A stability region approach, as described below, can help reasoning about the vehicle's safety.

Enforcing safety given human controls

Fig. 6 shows the plot of the stability region of the vehicle dynamics in the yaw rate ω_φ vs. the rear slip angle α_γ phase plane. The boundary between the stable and unstable region is the vehicle's operating limit for preserving stable and safe motion. The computation of the boundary is based on the vehicle and tire dynamics as well as human input [18]. If the car's motion under the human input commands and onboard sensing information is within the stable region, the embedded control will not interfere. Otherwise, the onboard vehicle controller will take action.

Given a stability region, a method can predict a state trajectory assuming constant control input. When a trajectory violates the stability region, a containment-preserving correction is applied. One option is to simply override the user's control input with the most similar non-infringing one. This relates to a rich literature in viability theory, such as [19, 20]. Highly-related problems are those of computing reachable sets for dynamical systems [21]. An alternative to overriding the user's control is to employ haptic feedback to negotiate with the user. Initially provide a guiding force, which hints at better courses of action, when a breach is distant. Then increase the corrective forces as the breach draws near [22].

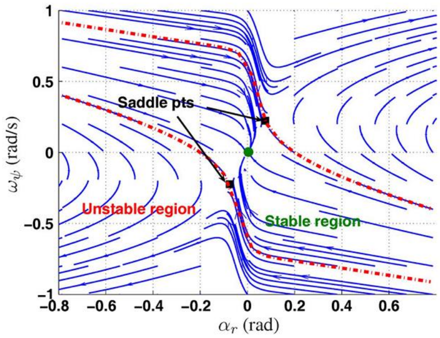

Figure 6 *The stability region in a car's yaw rate vs. slip angle phase portrait under zero tire slip ratios*

Security

Vehicles are subject to all sorts of well-publicized attacks. In this article, we focus on the following attacks uniquely targeted at remote driving: (i) A malicious attacker attempting to gain control of and inject commands into the networked control link between the car and the remote driving center; (ii) a DDoS attack against the vehicle; (iii) threats to the car itself, i.e., attempts by a malicious attacker to corrupt the firmware in the car so as to cause it to misbehave; (iv) malicious or corrupt remote drivers.

Secure communications channel

Establishing secure communication channels between the car and the driving center resists attempts by attackers to spoof connections to gain control of the car. Conversely, a malicious attacker should not be able to inject packets into the stream of sensor data going from the car to the driving center. A secure sockets layer (SSL) for all network communication is an immediate solution. During the SSL handshake, the car and the driving center authenticate each other using their public keys. SSL then establishes an encrypted communication channel, which defends against an attacker's attempts to spoof communication or inject packets.

DDoS attack

DDoS attack is carried out by surrounding vehicle(s) to jam the vehicular communication channel. The attack can be alleviated by proposed multi-channel and multi-path transmissions. Additional approaches such as randomized channel hopping within one or multiple types of wireless network can also be considered. In the worst case, where all possible channels are jammed, the car needs to rely on emergency autonomy discussed previously to stop and summon the in-vehicle driver.

Protecting the car

Malicious attackers may compromise the software running on the vehicle [23, 24]. Given the development of trusted hardware, such as the trusted platform module (TPM) [25], this problem can be avoided by attest the software stack executing on the car. For example, the TPM attests the BIOS and the bootloader of the car, which in turn attest the upper layers of software. The TPM provides a digitally signed set of such attestations to the remote driver. The driving center ensures the integrity of the car before transferring control of the remote vehicle to one of its employees. Since a modern vehicle consists of multiple subsystems, each of which is entrusted to a different set of controllers, the remotely driven car can hold a hierarchical attestation protocol. The driving center simply attests the main on-board computer. This on-board computer, in turn, will be entrusted to establish the trustworthiness of individual controllers.

Checking remote drivers

Malicious remote drivers may misuse their control of the vehicle to cause accidents. In principle, we entrust the remote driving center with the responsibility of vetting individual operators before letting them drive. Nevertheless, the control commands issued by the remote driver need to pass the collision avoidance check in the remotely driven car to avoid malicious controls. In addition, a computerized abnormality monitor can be developed to detect and disengage potential anomalous remote driver and notify the trust-able service provider.

Conclusions

This article examines the possibility to realize driving in a different direction from the current paradigm of on-board driving or the heavily studied purely AI-based approaches. Given progress in reliable wireless transmission, rapidly developing robotic control technology and sophisticated visualization tools, we believe that remote driving technology is likely to be achieved sooner than later. This will allow the proliferation of driverless car and will act as the necessary stepping-stone towards complete autonomy.

References

[1] C. Urmson. The latest chapter for the self-driving car: mastering city street driving, Apr. 2014.

[2] W. Knight. Driverless cars are further away than you think, October 2013.

[3] L. Yvkoff. Many car buyers show interest in autonomous car tech, April 2014.

[4] T. Fong, C. Thorpe, and C. Baur. Advanced interfaces for vehicle teleoperation: Collaborative control, sensor fusion displays, and remote driving tools.Autonomous Robots, 11(1):77–85, 2001.

[5] J. Guivant, S. Cossell, M. Whitty, and J. Katupitiya. Internet-based operation of autonomous robots: The role of data replication, compression, bandwidth allocation and visualization. Journal of Field Robotics, 29(5):793–818, 2012.

[6] A. Kelly, N. Chan, H. Herman, D. Huber, R. Meyers, P. Rander, R. Warner, J. Ziglar, and E. Capstick. Real-time photorealistic virtualized reality interface for remote mobile robot control. The International Journal of Robotics Research, 30(3):384–404, 2011.

[7] S. Gnatzig, F. Chucholowski, T. Tang, and M. Lienkamp. A system design for teleoperated road vehicles. InICINCO (2), pages 231–238, 2013.

[8] R. Stewart. Stream control transmission protocol. 2007

[9] A. Ford, Alan, C. Raiciu, M. Handley, and O. Bonaventure. TCP extensions for multipath operation with multiple addresses. Internet Engineering Task Force, RFC 6824 (2013).

[10] J. Cloud, F. Calmon, W. Zeng, G. Pau, L. Zeger, and M. Médard. Multi-path tcp with network coding for mobile devices in heterogeneous networks. In Vehicular Technology Conference (VTC Fall), 2013 IEEE 78th, pages 1–5. IEEE, 2013.

[11] V. Singh, S. Ahsan, and J. Ott. MPRTP: multipath considerations for real-time media. In Proceedings of the 4th ACM Multimedia Systems Conference, pages 190–201. ACM, 2013.

[12] A. Elqursh and A. Elgammal. Online moving camera background subtraction. In Proceedings of the European Conference on Computer Vision (ECCV), pages 228–241. Springer, 2012.

[13] A. Elqursh and A. Elgammal. Online motion segmentation using dynamic label propagation. In Proceedings of the International Conference on Computer Vision (ICCV). 2013.

[14] Y. Li, Z. Littlefield, and K. E. Bekris. Sparse Methods For Efficient Asymptotically Optimal Kinodynamic Planning. In Workshop on the Algorithmic Foundations of Robotics (WAFR), 2014.

[15] K. E. Bekris and L. E. Kavraki. Greedy but Safe Replanning under Kinodynamic Constraints. In IEEE Intl. Conf. on Robotics and Automation (ICRA), 2007.

[16] I. Abraham and J. Yi. Model predictive control of buoyancy-driven autonomous underwater gliders. Submitted to 2015 American Control Conference, 2015.

[17] C. E. Beal and J. C. Gerdes. Model predictive control for vehicle stabilization at the limits of handling. IEEE Trans. Contr. Syst. Technol., 21(4):1258–1269, 2013.

[18] J. Yi, J. Li, J. Lu, and Z. Liu. On the dynamic stability and agility of aggressive vehicle maneuvers: A pendulum-turn maneuver example.IEEE Trans. Contr. Syst. Technol., 20(3):663–676, 2012.

[19] J.-P. Aubin. Viability Theory. Systems and Control: Foundations and Applications. Nirkhauser, 1991.

[20] E. J. Rossetter, J. P. Switkes, and J. C. Gerdes. A gentle nudge towards safety: Experimental validation of the potential field driver assisstance system. InProceedings of the American Control Conference, 2003.

[21] B. Donald, P. Xavier, J. Canny, and J. Reif. Kinodyamic motion planning.Journal of ACM, 40(5):1048–1066, 1993.

[22] M. Steele and R. B. Gillespie. Shared control between human and machine: Using a haptic stering wheel to aid in land vehicle guidance. In 45th Annual Meeting of the Human Factors and Ergonomics Society, October 2001.

[23] W. Choi, A. M. Winer, and S. E. Paulson. Factors controlling pollutant plume length downwind of major roadways in nocturnal surface inversions. Atmospheric Chemistry and Physics, 14(13):6925–6940, 2014.

[24] K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, and S. Savage. Experimental security analysis of a modern automobile. In IEEE Symposium on Security and Privacy, 2010.

[25] R. Sailer, X. Zhang, T. Jaeger, and L. van Doorn. Design and implementation of a TCG-based integrity measurement architecture. In USENIX Security Symposium, 2004.

Ruilin Liu *(rl475@cs.rutgers.edu) received B.S. in computer science and techonology from Shandong University, China, in 2008, and M.S. degree from Tsinghua University, Beijing, China, in 2012. He is currently working towards his Ph.D. degree in computer science in Rutgers University. His research interest includes mobile and pervasive computing, vehicular networks, and urban computing.*

Kostas Bekris *(kostas.bekris@cs.rutgers.edu) works in robotics and is an Assistant Professor of Computer Science at Rutgers and a member of the CBIM center. He received a BS from the University of Crete, Greece and MS and PhD degrees at Rice University under the guidance of Prof. Lydia Kavraki, all in Computer Science. He was an Assistant Professor at the University of Nevada, Reno from 2008 to 2012. His interests include motion and task planning, manipulation and multi-robot systems. His group is supported by NSF, NASA, DHS, DoD and is affiliated with DIMACS and the DHS Center of Excellence CCICADA.*

Ahmed Elgammal *(elgammal@cs.rutgers.edu) is an associate professor at the Department of Computer Science, Rutgers, the State University of New Jersey. He is also a member of the CBIM center and CAIP at Rutgers. He received his Ph.D. degree in computer science from the University of Maryland, College Park, 2002. His primary research interest is computer vision and machine learning. His research focus includes human activity recognition, human motion analysis, tracking, human identification, and statistical methods for computer vision. He develops robust real-time algorithms to solve computer vision problems in areas such as visual surveillance, visual human-computer interaction, virtual reality, and multimedia applications. His interest also includes research on document image analysis.*

Vinod Ganapathy *(vinodg@cs.rutgers.edu) is an Associate Professor of Computer Science at Rutgers University. He received the B.Tech. degree in computer science and engineering from IIT Bombay, in 2001, and the Ph.D. degree in computer science from the University of WisconsinMadison, in 2007. His research interests are in computer security and privacy, software engineering, mobile systems, and virtualization.*

Mario Gerla *(gerla@cs.ucla.edu) is a Professor in the Computer Science Dept at UCLA. He holds an Engineering degree from Politecnico di Milano, Italy and the Ph.D. degree from UCLA. He became IEEE Fellow in 2002. At UCLA, he was part of the team that developed the early ARPANET protocols under the guidance of Prof. Leonard Kleinrock. He joined the UCLA Faculty in 1976. At UCLA he has designed network protocols for ad hoc wireless networks (ODMRP and CODECast) and Internet transport (TCP Westwood). He has lead the ONR MINUTEMAN project, designing the next generation scalable airborne Internet for tactical and homeland defense scenarios. His team is developing a Vehicular Testbed for safe navigation, content distribution, urban sensing and intelligent transport. He serves on the IEEE TON Scientific Advisory Board. He was recently recognized with the annual MILCOM Technical Contribution Award (2011) and the IEEE Ad Hoc and Sensor Network Society Achievement Award (2011).*

Liviu Iftode *(iftode@cs.rutgers.edu) is a Professor of Computer Science at Rutgers University. He received the Ph.D. degree in computer science from Princeton University, in 1998. His research interests are in operating systems, distributed systems, systems security, mobile and pervasive computing, and vehicular computing and Networking.*

Melchi Michel *(melchi.michel@rutgers.edu) received his Ph.D. from the Department of Brain and Cognitive Sciences at the University of Rochester in 2007. He was a postdoctoral fellow in the Center for Perceptual Systems at the University of Texas at Austin from 2007-2012 and joined the Rutgers faculty in 2012. I study the human visual system with a focus on investigating how we integrate sensory information to make perceptual judgments, how we exploit statistical regularities in the environment, and how we adapt when these statistical regularities are altered or when new statistical contingencies are introduced. My research addresses a broad spectrum of topics ranging from basic questions about visual signal detection, visual search, perceptual learning, and neural coding to potential clinical applications in assessing the visual performance of retinopathic patients and radiologists and evaluating the potential impact of training interventions.*

Jingang Yi *(jgyi@rutgers.edu) is an Associate Professor of Mechanical and Aerospace Engineering and a Graduate Faculty member of Electrical and Computer Engineering at Rutgers University. He received the B.S. degree from Zhejiang University, China, the M.Eng. degree from Tsinghua University, China, and the M.A. and Ph.D. degrees from the University of California, Berkeley. Dr. Yi's research interests include autonomous robotic systems, dynamic systems and control, mechatronics, automation science and engineering, with applications to and transportation systems, biomedical systems, civil infrastructures. Dr. Yi is a member of American Society of Mechanical Engineers (ASME) and a senior member of IEEE. He was a recipient of the 2010 US NSF CAREER Award.*