

Autonomous driving in urban environments: approaches, lessons and challenges

BY MARK CAMPBELL¹, MAGNUS EGERSTEDT², JONATHAN P. HOW³
AND RICHARD M. MURRAY^{4,*}

¹*Cornell University, Ithaca, NY 14853, USA*

²*Georgia Institute of Technology, Atlanta, GA 30332, USA*

³*Massachusetts Institute of Technology, Cambridge, MA 02139, USA*

⁴*California Institute of Technology, Pasadena, CA 91125, USA*

The development of autonomous vehicles for urban driving has seen rapid progress in the past 30 years. This paper provides a summary of the current state of the art in autonomous driving in urban environments, based primarily on the experiences of the authors in the 2007 DARPA Urban Challenge (DUC). The paper briefly summarizes the approaches that different teams used in the DUC, with the goal of describing some of the challenges that the teams faced in driving in urban environments. The paper also highlights the long-term research challenges that must be overcome in order to enable autonomous driving and points to opportunities for new technologies to be applied in improving vehicle safety, exploiting intelligent road infrastructure and enabling robotic vehicles operating in human environments.

Keywords: autonomous driving; robotics; intelligent transportation

1. Introduction

The field of autonomous driving has a rich history, with early demonstrations in the 1980s and 1990s showing the viability of building cars that can control their own motion in complex environments such as highways and urban streets (see Dickmanns (2002) and Thorpe *et al.* (1988) for excellent reviews of this early work). Advances in sensing and computation technologies have further spurred interest in autonomous driving and many advances have been made in the past decade, as evidenced, for example, by the series of competitions sponsored by the US Department of Defense (Defense Advanced Research Projects Agency (DARPA) 2007). In this article, we attempt to summarize the current state of the art, describe some of the major research challenges that remain to be solved and indicate some of the opportunities for the application of these exciting technologies.

Modern autonomous vehicles are capable of sensing their local environment, classifying the types of objects that they detect, reasoning about the evolution of the environment and planning complex motions that obey the relevant rules

*Author for correspondence (murray@cds.caltech.edu).

One contribution of 10 to a Theme Issue ‘Traffic jams: dynamics and control’.

of the road. Although less well developed, substantial progress has also been made in reacting to ‘unexpected’ situations where failures can occur in the vehicle systems or agents in the external environment do not behave as predicted by internal models. The ability to navigate autonomously in these complex situations is accomplished by combining a variety of technologies from different disciplines that span computer science, electrical engineering, robotics and controls, to name a few. Demonstration vehicles such as those developed by Dickmanns and his group (Dickmanns 2002), the CMU Navlab (Thorpe *et al.* 1988; Sukthankar *et al.* 1997) and the DARPA Grand Challenge (DGC) vehicles have all succeeded in combining these technologies into working systems.

As one recent example of the state of the art in autonomous driving, the three DGC competitions, held in March 2004, October 2005 and November 2007, involved hundreds of autonomous vehicles that demonstrated varying levels of capability. In the first and second competitions, vehicles had to demonstrate their ability to navigate in an off-road, desert environment, travelling up to 150 miles (240 km) at speeds of up to 50 mph (80 km h⁻¹) using sensory information to determine where it was safe to drive. In the first competition, only five vehicles travelled more than a mile, with the vehicle travelling the farthest only travelling approximately 12 km (7 miles). By the second competition, three vehicles were able to complete 212 km (132 miles) of driving with no human input. The third round of the competition shifted to urban driving, with six vehicles completing approximately 60 miles of driving in environments that included city streets, parking lots, traffic intersections and both human and robotically controlled vehicles.

The contributions of this paper are to give a brief overview of some of the main technologies and architectures that have been used to achieve autonomous driving, describe some of the lessons that we have learned in implementing autonomous navigation systems and, most importantly, describe some of the upcoming challenges and opportunities that we see for the field. Our insights are motivated by our participation in the DGC competition, as well as our research interests in autonomy, robotics and control. We focus specifically on autonomous vehicles operating in traffic and touch on some of the interesting dynamics that arise from interactions between vehicles.

The paper begins with a more detailed description of the 2007 DARPA Urban Challenge (DUC) competition, focusing on the main approaches, architectures and lessons from the competition. We then describe some of the challenges facing autonomous driving, especially those related to driving in environments with traffic and humans. Finally, we provide a description of some of the opportunities for autonomous vehicle technologies in the short and medium term.

2. Approaches and lessons from the 2007 DUC

We begin with a brief summary of the 2007 DUC, focused on the approaches taken by the authors’ teams and some of the experiences and lessons based on our participation in the competition.

(a) Urban challenge overview

The DUC was announced in June 2006 as the third in the DGC series of competitions. While the first two DGC competitions focused on driving in off-road (desert) environments, the third DGC competition was based on urban driving. At the time of the initial announcement, a set of basic driving scenarios was described, including navigation on roads with other traffic, operation at intersections with precedence rules and parking in parking lots. To participate in the final race, teams had to pass a site visit in April 2007 that tested basic autonomous functionality and a qualification event in October 2007 (just before the race).

The basic specifications for all of the DGC competitions were that the vehicles had to operate completely autonomously, with no human input. A route network definition file (RNDF), which included GPS locations for intersections and selected waypoints, plus specifications of the number of lanes and directions on road segments, was provided ahead of time, with a description of the tasks that needed to be carried out. In the case of the urban challenge, the tasks were given as individual ‘missions’ that consisted of reaching a set of checkpoints in a certain order. The vehicles had to plan their routes to the checkpoints using available roads, including the ability to re-plan if the route was blocked. GPS positioning was available, although signals could be blocked by buildings or trees. The only other input into the vehicle was a ‘E-Stop’ unit that signalled whether the vehicle was allowed to run autonomously and allowed the race organizers to pause or disable a vehicle remotely.

The site visits consisted of a set of six basic tests that included safety demonstrations, lane following, mission planning and intersection operations. The safety demonstration consisted of bringing the vehicle to a speed of 32 km h^{-1} (20 mph) and then showing that it could be brought to a stop within a distance of 20 m using the remote E-Stop unit. Additional tests were used to verify that the vehicle could travel down a road with marked lane lines, stop at intersections, interact with stopped and moving vehicles on the road (going in either direction) and properly operate at intersections with a variety of patterns of arrivals of other vehicles. A typical site visit course is shown in [figure 1a](#).

The qualifying event consisted of three separate tests, in addition to a basic vehicle inspection and safety verification (stopping using the E-Stop unit). The qualifying event was carried out at George Air Force Base in southern California, an unused military base consisting of suburban streets and buildings. An overview of the qualification event courses is shown in [figure 1b–d](#). Area A ([figure 1b](#)) tested the ability to operate in traffic and consisted of a road with approximately 12 cars circling in opposite directions at $16\text{--}20 \text{ km h}^{-1}$ (10–15 mph) around a loop. The autonomous vehicles had to cross the oncoming traffic to turn into and out of a road across the middle of the course, obeying appropriate traffic rules. Area B ([figure 1c](#)) was a long course that was used to test the ability to plan and execute missions on different types of roads, navigate and park in parking lots and avoid static obstacles, including parked cars and areas marked off by traffic cones. Area C ([figure 1d](#)) was an intersection test, with a sequence of increasingly complicated tests involving different combinations of cars arriving at the intersection before and after the autonomous vehicle.

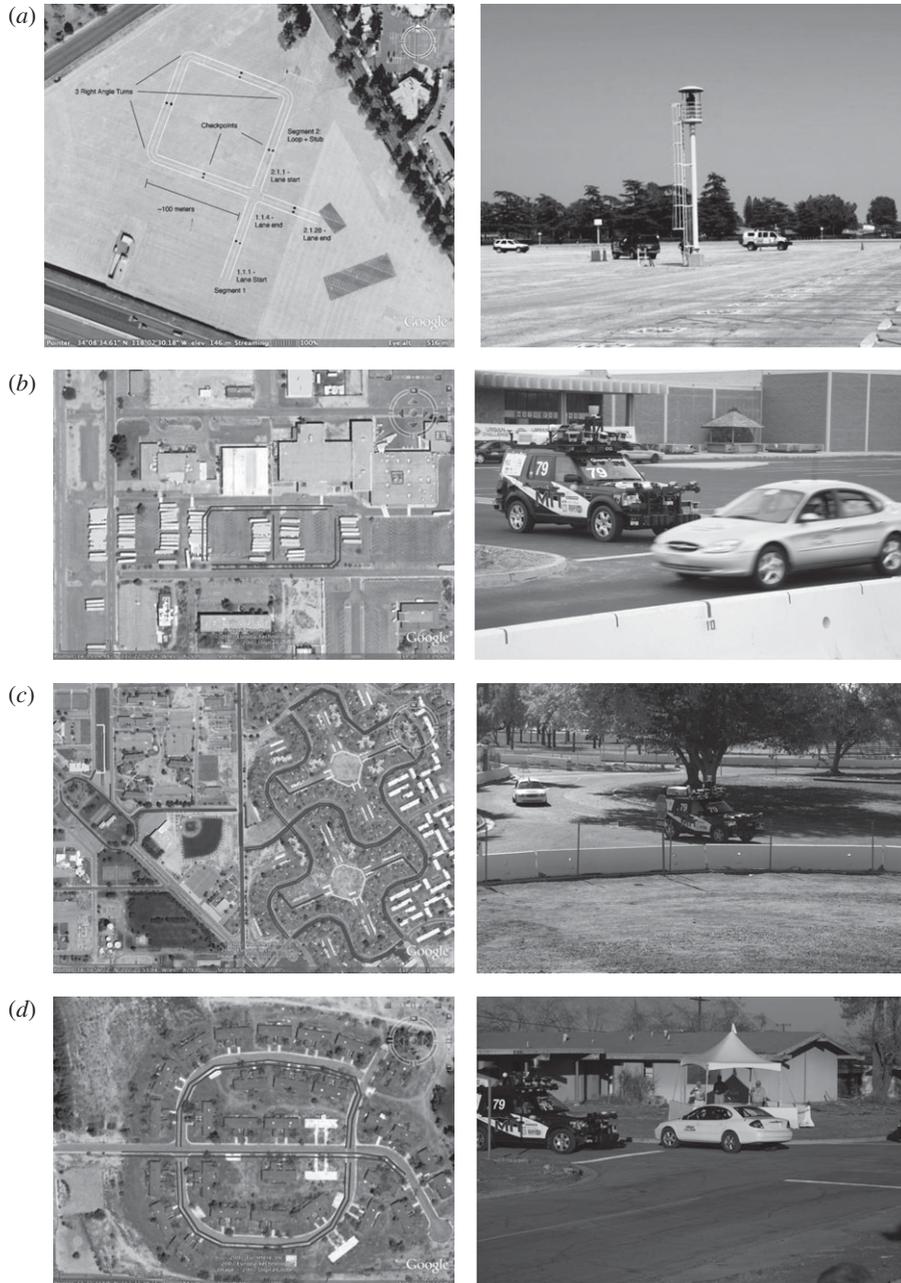


Figure 1. Overview of the site visit and NQE courses used for the DUC. (a) Site visit course, (b) NQE area A, (c) NQE area B and (d) NQE area C.

Of the 35 teams that participated in the national qualifying event (NQE), only 11 teams continued on to the final urban challenge event (UCE). The course is shown in figure 2, and it included mostly paved roads with one lane in each direction. It also included parking spaces, a traffic circle, a one way dirt road, and a four lane separated highway. Each vehicle was required to complete three



Figure 2. The urban challenge event. (a) Line-up of the 11 cars in the final race. (b) Course map with several environmental elements noted.

missions, each approximately 30 km long. The individual missions included six sub-missions which required the cars to achieve a series of checkpoints and return to the traffic circle area. All 11 qualifying robots were allowed to interact in the UCE course simultaneously with additional human driven cars.

Of the 11 final teams, only six autonomous vehicles finished all required missions. The first prize was awarded to 'Boss' developed by Tartan Racing, a team led by Carnegie Mellon University. The second-place finisher was Stanford Racing Team's 'Junior'. In third place was 'Odin' from Team VictorTango at Virginia Tech. Vehicles from MIT, Cornell, and the University of Pennsylvania/Lehigh also successfully completed the course. Team Oshkosh, IVS/Honeywell, Team UCF, Team AnnieWay, and CarOLO started the race but did not finish.

(b) System architectures

A variety of system architectures were used by the teams that participated in the DGC competitions. In this section we give a brief overview of the main approaches that were used and some of the trade-offs involved. Much more detailed descriptions of the architectures of the DGC vehicles are available in the literature (Iagnemma & Buehler 2006; Iagnemma *et al.* 2008).

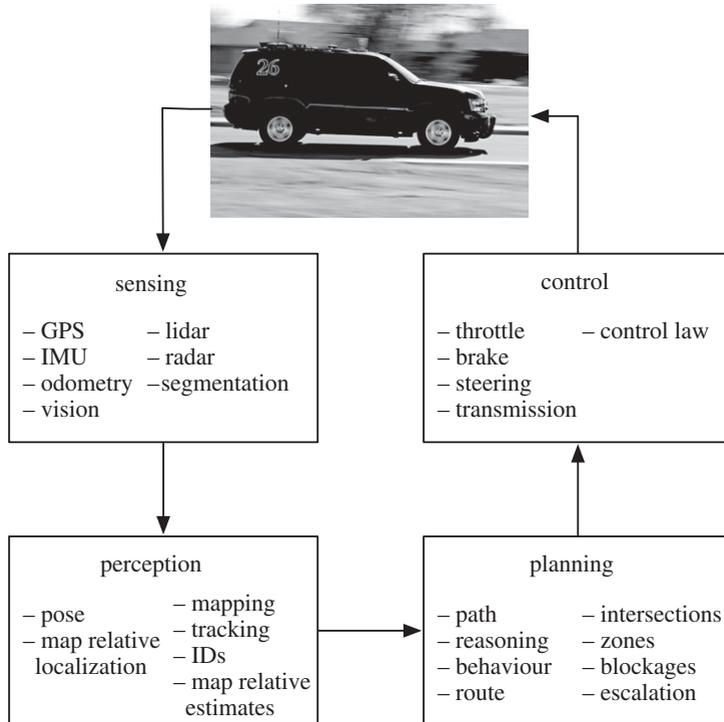


Figure 3. High level systems architecture for urban driving.

At a high level, most vehicles in the competition decomposed the problem into four basic subsystems: sensing, perception, planning and control. A sample block diagram showing these functions is shown in figure 3.

The sensing subsystem was responsible for taking raw data measurements. For the vehicle, this included GPS, IMU (inertial measurement unit) and odometry measurements (or an off the shelf system that fused these together); several teams also included vision for lane and stop line detection. For perceiving the static and dynamic urban environment, measurements included laser range finders, radar and cameras. Most teams also segmented the laser (e.g. clustering) and vision data (e.g. lane finding) in order to produce a data product of smaller size that was easier to process.

The perception subsystem was responsible for creating usable information about the vehicle and its environment. Vehicle estimation included pose (inertial position, velocity, attitude, rates) as well as map relative information (e.g. the vehicle location within a lane or map); the latter typically used vision or laser measurements to help to produce map relative estimates. Estimation of the environment was accomplished in a number of ways, primarily because of the variations in sensors, computation and resources. For example, most teams developed monitors around the vehicle (front right, front left, side, etc.) and estimated/reasoned about elements such as the location, velocity, lane of other cars, and sensor occlusions. For example, the Cornell team developed a Rao-Blackwellized particle filter (Miller & Campbell 2007) to accurately track the

other cars relative to its vehicle with sub-metre accuracy. Additional reasoning components were then developed using these data to provide the planners with appropriate information.

The planning subsystem typically included common components such as path planners, behavioural planners and route (map) planners. These varied, however, across different implementations. For example, the MIT team used a combination of a navigator that specifies a goal point and a motion planner that finds a feasible path to the goal using a unique sampled-based approach called closed-loop rapidly exploring random tree (Kuwata *et al.* 2009), while the Cornell and Caltech teams used optimization based path planners (Hardy *et al.* 2008). Most behavioural planners were built around finite-state machine logic, although the Georgia Tech team used a variety of low-level behaviours and selected based on either high-level decision-making or lower-level ‘voting’. A key element in most planners was reasoning about the probabilistic information coming from the perception subsystem, which was typically accomplished with a finite-state machine. For special behaviours, such as operation at intersections, zones, and blockages, custom components were usually designed. Finally, most teams developed an ‘escalation planner’ for anomalies that could occur (such as large perception mistakes); if this was invoked, a variety of actions were taken, typically after timers expired, including reducing or removing constraints in planners.

Finally, the control subsystem included the actual actuators and commands to drive the car; information for the control law would come from some combination of the higher level planning (i.e. the proposed path), and/or direct sensing in some cases in order to increase the speed of response and avoid obstacles. The vehicle actuation varied across the cars and the teams. Smaller cars, such as the Prius and Jetta, made use of existing drive by wire systems, while larger cars, such as the Chevrolet Tahoe and Range Rover, typically used retrofitted motor drives for actuators such as braking and steering. (Each team’s choice of vehicle was based on their assessment of the probable manoeuvres required, available resources and partnerships, and ease of modification.)

(c) Competition experiences

The architectures and approaches that were developed by each of the teams were tested at three stages during the DUC competition: a site visit approximately six months before the competition, the NQE in the week before the competition and, for those teams that qualified, the final competition, on 3 November 2007. In this section we briefly describe how the four team represented by this paper fared at the NQE and competition.

At the NQE, the Georgia Tech vehicle (shown in figure 4) first attempted area C. The run was fairly successful, albeit a bit cautious and slow as the vehicle was treating obstructions on the side of the road (e.g. barrels) as parked vehicles that require a full-fledged lane-changing manoeuvre for their negotiation. The next test, area A, was less successful in that a GPS failure caused the vehicle to believe that it was stationary, while in reality it was driving straight into a concrete rail. The car was quickly repaired and moved to area B. There, the car did not go far since one result of the crash was that all the sensors were uncalibrated. A night of calibration and programming left the car ready for area C again, which it cleared without problems. However, the two failed



Figure 4. The Georgia Tech vehicle Sting 1 at NQE test area C.

attempts (areas A and B) the previous day caused the Georgia Tech team to be taken out of the competition. Both of these failed attempts can to a certain degree be attributed to the complexity of the system, with unforeseen error modes and connections between modules having a negative impact on the overall performance of the system.

Caltech's vehicle, Alice, encountered several unexpected issues in its testing. One set of problems arose from the tight spacing at several points in the course, where concrete barriers and other obstacles did not allow Alice to traverse a section of road while maintaining the required 1 m spacing to each side of the vehicle. In order to proceed through the area, Alice had to progress through a series of internal planning failures before finally driving with reduced buffers on each side of the vehicle, causing very slow (but mostly correct) progress. This was fixed in subsequent runs by reducing the size of the buffers that Alice maintained on its sides. A more serious bug was detected in area B, the traffic merging test, related to the way that intersections were handled by the planning logic. While the intersection handling logic was active, another part of the higher-level logic planner could switch into a new state if it detected a nearby vehicle (e.g. one of the human-driven cars was predicted to collide with Alice or its buffer region). This change in state de-activated the intersection handler and could cause the vehicle to enter the intersection when the path became clear (without invoking the proper merge logic). This bug was very difficult to fix (and test) in the field and eventually caused Alice to execute some very unsafe behaviour on the second attempt at area A, and it was disqualified from the competition. Alice was able to successfully complete area B on a second attempt, as well as almost all of area C.

The MIT NQE experience started well in area B, but then had some issues in area A (miscalculation of the time gap between oncoming cars led to a very conservative behaviour at one of the intersections) and area C (executed the precedence checking at the intersections very well, but went around an added road block rather than performing a U-turn as desired). The area A issues were resolved on the second attempt, but, even though Talos was still progressing, it ran out of time on the second attempt of area B (took a long detour) and went off-road again at the first blockage in area C. Fortunately these problems were

resolved, and areas B and C were completed on the third attempt, and MIT was included in the UCE. The final event went well, and despite several incidents with other robots (discussed later), Team MIT finished fourth.

The Cornell NQE experience included one flawless run in area C, and two runs in both areas A and B. In area A, Cornell had a conservative time gap for merging in the first run, which caused the vehicle to complete only several loops of the course; this time gap was then reduced, and nearly 10 loops were completed in the second run. In first area B run, the primary issue Cornell had was driving through the ‘Gauntlet’ area, with parked cars on both sides. Because of the large size of the Chevrolet Tahoe from Cornell, Cornell had to move over the centreline; however, Cornell’s path planner had a constraint that did not allow this, so a U-turn manoeuvre was started. After the initial area B run, Team Cornell slightly adjusted the centreline spacing parameters in the online path planning optimization, which allowed Cornell to successfully complete area B during the second run. In the UCE, Cornell was one of six vehicles to finish the challenge. Several incidents are discussed later, but the primary performance item of note was a faulty throttle during the final mission that capped the speed between 10 and 15 km h⁻¹.

Many other issues affected the overall performance of the vehicles, including sensor noise, environmental complexity and other sources of uncertainty. Each team chose specific approaches to deal with these sources of potential error, including failures in both sensing and actuation systems. More details are available in technical papers and reports about the individual vehicles (Burdick *et al.* 2007; Leonard *et al.* 2008; Miller *et al.* 2008).

The 2007 DUC was won by Carnegie Mellon University (Urmson *et al.* 2008), with second place going to Stanford (Montemerlo *et al.* 2008) and third place to Virginia Tech (Bacha *et al.* 2008). The other team to finish the course was the University of Pennsylvania/Lehigh (Bohren *et al.* 2008).

3. Challenges in autonomous driving

In this section, we describe some of the future research directions that remain open challenges in order to integrate autonomous vehicles into existing road and transportation networks. We have broken this down into six overlapping areas where we believe the most research is required.

(a) Systems integration

Developing robust autonomous vehicles poses several integration challenges. There are numerous constraints on the navigation system that must be addressed, such as the location, field of view, and mounting type for the sensors, and cooling and power for the computer system. Many of the DGC vehicles had numerous sensors for full coverage, but a key challenge for the future is to determine how autonomous driving can be accomplished with fewer, cheaper and conformal mounted (thus limited field of view) sensors.

Software integration was challenging because of the large number of sensor processing, navigation, planning, control, and safety processes that must run simultaneously. Enabling asynchronous operations using tools such as MIT’s lightweight communications and marshalling (Huang *et al.* 2009) proved to be

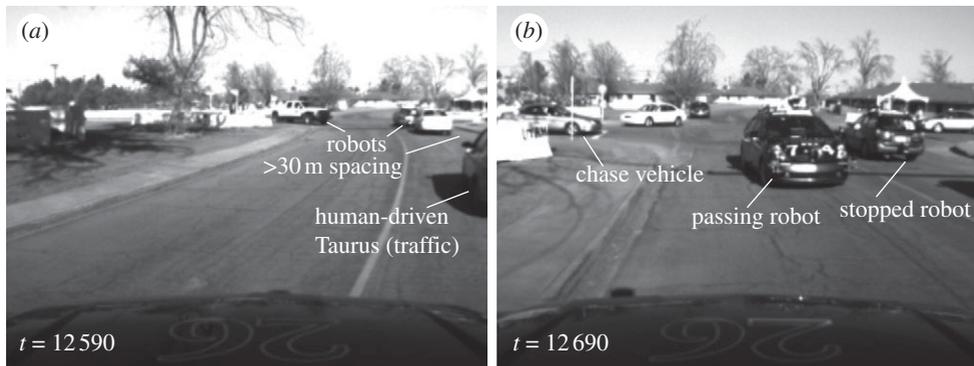


Figure 5. Robots in a traffic jam: view from Team Cornell's vehicle (as a driver would see). (a) Team Cornell's vehicle approaches the traffic jam, and begins to pass a stopped vehicle deemed disabled. (b) Team Cornell's car moves into the opposite lane to make a pass, but gets trapped in the wrong lane as the 30 m gap closes. Other vehicles pass Team Cornell's car as it waits for the intersection to clear.

very beneficial. Efficiently handling the load on each processor can also be difficult. A further challenge is to ensure that the integration of the hardware and software is done in such a way as to ensure sufficient robustness to component (e.g. sensor or computer) failures.

Algorithmic integration issues include ensuring that the assumptions are consistent between the algorithms on what (and the precision/noise levels) information they are either providing or receiving. This is typically handled using interface requirements documents, but this was difficult to manage on the time scales of the DGC, and can also be difficult to predict ahead of time. Testing of the overall system then becomes paramount to ensure that the overall performance and robustness meet the specifications.

Determining what information is relevant, and how to display it to any human occupants or remote supervisors is a further challenge that also needs to be addressed for an autonomous system to be readily adopted.

(b) Prediction and trust

The recent DUC, despite its enormous success, demonstrated the current brittleness of robotic intelligence, where small perception mistakes would propagate into planners, causing near misses, human-assisted restarts, and even a few small accidents. Fletcher *et al.* (2008) give a detailed technical account of collision between MIT and Cornell, which could have been prevented if either of the cars had anticipated the actions of the other car. As another example, consider a traffic jam scenario from the DUC (Miller *et al.* 2008). Figure 5a shows Team Cornell's car approaching a traffic jam: a robot was disabled at the stop line, with two human driven cars behind it. The last of the vehicles left a large space between it and the other two cars. Unfortunately, this large space was longer than the 30 m DARPA-specified safety zone (i.e. 'rule') that surrounds each intersection. Because the vehicle was outside the safety zone, Team Cornell's car was technically allowed to consider it disabled and pass, according to the rules.

After waiting 10 s, Team Cornell's vehicle did exactly that, as shown in figure 5*b*. But, a human driver, with learned behaviours and practice, would never take this action because they would 'anticipate' that a car would potentially make a turn, and move towards our vehicle in the opposite lane. This is also exactly what happened, as shown in figure 5*b*.

The traffic jam event is unique because no DUC rules were broken: robots were allowed to pass other disabled robots after waiting for 10 s, provided they were outside intersection safety zones. Despite following the rules, the decisions made by the vehicle were still undesirable. Such a situation emphasizes the importance of higher level reasoning about other vehicles' behaviour that was not incorporated into the current state of the art: had Team Cornell's car anticipated the motions of the other vehicles, it would never have tried to pass in the first place.

The analysis of these events generally points to the difficulties in connecting the currently separate modules for probabilistic perception and deterministic planning. The current state of the art typically consists of separate approaches to perception and planning, with relatively little work on integrated intelligent autonomous systems, much less on the development of a unified, formally verifiable theoretical framework. The work of Rosencrantz *et al.* (2003), Brooks *et al.* (2006) and Gonzalez & Stentz (2007) used partially observable Markov decision processes for path planning under uncertainty. But, while the problem is cast in a formal way, these methods generally require discretization that does not scale well; recent results have attempted to overcome these challenges (Pineau *et al.* 2006).

Before beginning to address anticipating the actions of other vehicles, it is clear that more reliable perception methods are required. The current state of the art of tracking from a moving vehicle was in the DUC; Team Cornell developed a formal Bayesian estimator that could track car-like obstacles reliably for tens of seconds (Miller & Campbell 2007), to less than 1 m, with less than 0.2 m s^{-1} accuracy. But, on occasion, a cement barrier would be mistaken for a car, or a large bush would be initialized with a strange velocity because of occlusion reasoning. It is clear that more complex representations and reliability for perception are required. For example, including object identification information in the estimator, and selectively incorporating vision sensing could improve these mistakes. Additional research is also required to handle more complex scenes, with clutter, and obstacles such as people and bicycles. Improved ground models, representations and reasoning about sensor returns are potential areas of research to address these more complex environments. Finally, with reliable tracking and object classification of tens of seconds or more, one can begin to address the concept of estimating the behaviour of an object. Human drivers are excellent reasoners about other drivers' behaviours, based on their experiences, even with a small amount of data (eye contact). Areas of research to address this problem include hierarchical graph models that build upon the reliable, lower level tracking, and representations to capture the required higher level information for planning.

Given a robust, probabilistic snapshot of the environment, which includes diverse elements such as tracking, classification and behaviours, one can begin to consider the concept of anticipating the actions of others, or reasoning about what others may do, and use this information in planning. One key question is, what level of abstract is required for anticipation? Perhaps it is just

position/velocity/shape, which would be helpful in preventing collisions. But, object identity may give more information, such as reasoning about the potential motion differences between a person and a telephone pole. Finally, human drivers still behave in a way that yields an important goal for research: they can typically tell the difference between a good or dangerous (or unaware) driver, and can use this information in their own decision making. Research here would require estimating the behaviours of other drivers, anticipating their potential actions, and incorporating this (probabilistic) information into planning—no small task.

Two other issues are worth noting when considering perception and prediction in intelligent vehicles. The first is the concept of trust. When driving past another car on the road, a car may come within 1 m of the other car. This is possible because the drivers both obey certain ‘rules of the road’; just as importantly, both also ‘trust’ that the other will obey these rules. In the DUC, teams incorporated trust in various ways, typically by assuming basic rules of the road. But, when these rules were breached, as in the MIT–Cornell collision (Fletcher *et al.* 2008), there were no mechanisms for compensation in order to prevent problems. Areas of research here include what representations to use for trust, and how to incorporate trust into both the perception and planning of the vehicle. The second issue is sensing range, as discussed in §3*e*.

While the DUC demonstrated that autonomous driving will require more robust and richer perception outputs of the environment, including anticipating and reasoning about other obstacles, planners must also be able to handle uncertainty, as the outputs will always be uncertain. This subject is described in more detail in the sections below.

(c) Interactions between agents

Research on improved sensing, control and decision algorithms will undoubtedly make autonomous driving more reliable in the future. However, increases in capabilities and infrastructure for supporting inter-vehicle communications will also play an important role. With the increased inter-vehicle communications, new types of algorithms will be possible that not only rely on the individually perceived world, but also on the shared experience between vehicles. Examples where this increased communications capability can be potentially useful include (i) by sharing information about driving decisions and intentions, a more safe coexistence between autonomous vehicles should be possible, (ii) the available information can be propagated through the network of cars to avoid hazards/obstructions/inconveniences such as accidents, traffic jams and ice-patches, and (iii) the fuel economy of the overall traffic network can be improved (in conjunction with smarter fuel systems) by propagating information about driving conditions. We will here discuss some of these issues.

In order for autonomous driving to reach its full potential, it is vitally important that the cars cooperate in the sense that they agree on traffic rules, whose turn it is to drive through an intersection, and so forth. For this, robust agreement protocols must be developed. Recent work on how to make multiple vehicles agree on common state variables, e.g. using consensus or gossip algorithm (Boyd *et al.* 2006; Olfati-Saber *et al.* 2007), provides a promising starting point for this undertaking.

When running such agreement algorithms, it is conceivable that not all vehicles will cooperate. They may, for example, be faulty, or simply driven by human operators, and such vehicles must be identified and isolated in order to balance autonomy with human inputs. This will be true on individual cars, but even more so in mixed human–robot networks. Questions of particular importance (that will have to be resolved using the available interconnections) include the following.

- (i) *Safety*: autonomous cars must be able to identify human-driven cars and then not drive into them even though they may violate the robot driving protocol.
- (ii) *Opportunism on behalf of the human drivers*: people are already driving badly on the road when the other cars are driven by people. How will they act if no-one is driving? This needs to be taken into account by the autonomous cars (i.e. not only will people not follow the ‘correct’ protocol—they might be outright hostile).
- (iii) *Collaborative versus non-collaborative driving*: how should non-cooperative vehicles be handled in an algorithmically safe yet equitable manner?

It should also be noted that humans rely almost entirely on non-verbal communication cues when signalling intent in traffic. An additional challenge when understanding how to structure inter-vehicle interactions is thus to try to capture this non-verbal communications aspect. For instance, the vehicles must be able to estimate which mode of operation neighbouring cars are in (not just their physical states such as positions and velocities) in order to be able to adequately predict and plan for upcoming manoeuvres.

To facilitate inter-vehicle communications and cooperation, it is important that an effective information-exchange infrastructure is in place. In fact, a vehicle-to-vehicle (V2V) communication technology is already under development (together with V2I—vehicle-to-infrastructure) as part of the Vehicle Infrastructure Integration initiative. The basic idea is to equip the vehicles with both transmission and reception capabilities, and to have them transmit automatically over the established ad hoc networks, e.g. based on short-distance wireless radio channels.

Some of the proposed uses of the V2V and V2I networks include using shared information for ensuring safe operation, avoiding traffic congestion and enabling more fuel efficient modes of operation (for a representative sample, see Bana & Varaiya 2001; Wu *et al.* 2004; Santa *et al.* 2008; Economist 2009).

In fact, using the V2V communications networks, information about pile-ups, traffic congestion, dangerous driving conditions, and so forth, will be propagated to vehicles, and the vehicles themselves will act as routers for enabling the needed multi-hop capabilities. In conjunction with this, network routers will be placed along main traffic arteries. The fact that such networks are already in the pipeline has implications for autonomous driving, in that certain aspects of the needed infrastructure are both economically and politically within reach.

(d) *Learning*

The DUC also demonstrated the limitations of robot cars to robustly learn from prior experiences and/or mistakes, which is a key characteristic for long-term operations envisaged for the future. For example, during the NQE, many teams discovered corners in the RNDF files, in particular near intersections, that caused difficulties for the route planners and the RNDF files had to be manually tweaked before the missions. Furthermore, few, if any, teams updated the RNDF

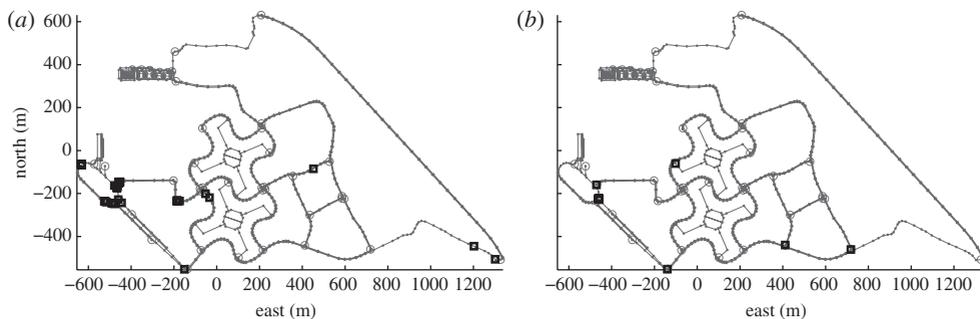


Figure 6. (a) Location of Cornell's vehicle (black square) during the 53 emergency brake slams it performed during the DUC. (b) Location of Cornell's vehicle (black square) during the 10 times it went into blockage recovery during the DUC.

database to re-align the road network in real time. For example, the MIT team used a system that was designed to use locally perceived information in preference to potentially inaccurate map data to navigate a road network while obeying the road rules. However, during the course of the UCE and NQE, many of the roads were traversed multiple times (10–20 for some), providing ample data to update the prior map information (encoded in the RNDF). For example, had there been time to implement this type of update it might have been particularly useful during the multiple slow descents of the MIT vehicle in the dirt road section. Some teams implemented a basic form of this adaptation using online updates of the safe speed to drive and the time estimates to traverse sections of the road network.

With the breadth of sensor data, from which important driving decisions must be made, learning from perception mistakes is a particularly important area. As an example, consider figure 6, which shows the DUC course (map), along with an overlay of 53 instances of emergency brake slams (figure 6a) during the final event by Team Cornell's vehicle, along with 10 instances of blockage recovery (i.e. on-line debugging; figure 6b). What is significant is not the number of these instances, but their repeatability over multiple passes near the same areas. Many of the commands occurred where a single concrete barrier jugged out from the others, making it appear (to the perception algorithms) that it was another car. Team Cornell's car went through the same brake/debug procedure at every pass in order to be safe; however, a human would have learned over time that the mistake (and subsequent plan) was repeatable, and could be learned.

Team MIT also had several incidents with other slowly moving vehicles that were incorrectly classified as static. Some approaches to address this issue appear in the literature, but these were thought to be too difficult with the complex sensor suite and scene geometries in the DUC, so these were not pursued and a simpler classification approach based on a minimum speed threshold was used. This approach worked well in testing and during the NQE, but complications appeared during the UCE during the numerous interactions with other autonomous vehicles. These complications included the collision with Cornell (Fletcher *et al.* 2008). Since these mistakes were repeated several times during the UCE for similar reasons, it is probable that a human operator would have learned from these mistakes, and thus it is plausible that a correction to this behaviour could

have been learned autonomously. However, there clearly is a trade-off in safety versus conservatism, and using learning algorithms in situations such as this where failure is not acceptable is a significant algorithmic challenge.

These examples illustrate some of the challenges for future autonomous driving, which includes the need to improve the ability of the robots to learn and adapt to the environment and learn from past experiences at all levels of the planning architecture.

For example, at a low level this includes learning about the vehicle and core system capabilities, such as the vehicle dynamics and how these might change over time, and in particular adapting to hardware or sensor failures (e.g. the stuck accelerator for Team Cornell) or calibration changes (Bohren *et al.* 2008). As mentioned above, it also includes detecting features/properties (e.g. road, curb, painted lines, sign locations, road surface quality and type) during initial visits, updating the database, and using that information in later trips. It also includes learning how to better execute advanced driving skills, such as judging which lane to be in when driving in heavy traffic, and perhaps recognizing where typical traffic delays occur and designing more efficient routes at certain times of the day.

Persistent, long-term operations will also be required at the higher level. For example, each team typically had a mechanism for freeing up the vehicle if no progress was being made (called an ‘escalation planner’ by Team Cornell, and ‘failsafe modes’ by Team MIT). These typically involved a relaxation of the constraints imposed on the planner as the stopped time increased. These constraints were typically tested extensively in simulation and experiment, but still occasionally led to some undesirable behaviours. For example, the failsafe timer relaxed the constraints posed by the roadside curbs when little progress in front of the full road width blockage added at the end of the test in the NQE area C. The result was that MIT Talos was able to complete the course by going around the roadblock instead of doing a U-turn. The U-turn behaviour was added, but then that caused a detour in a following area B test. More extensive simulation could be used to identify these types of behaviours, but this is difficult to do since any change in the planning, perception, or control systems can lead to a new, unexpected response. Thus, more formal verification and validation procedures are needed for future autonomous systems (see §3*f*). Furthermore, even though these algorithms were tuned extensively, the process for relaxing the constraints was typically static once the race started. A future challenge is to develop more systematic methods for designing the failsafe modes and enabling the mode switching laws to adapt to the behaviour during previous ‘unsticking’ events.

Trust was discussed previously as a key issue in perception. It is important that techniques also be developed to adapt the models/trust of other drivers, bikers, and pedestrians based on a classification of prior meetings. Classification models of the far-away behaviour can then be updated based on the perceived levels of safety of recent meeting events (e.g. at intersections and crosswalks).

(e) *Scaling up*

As autonomous driving transitions from the somewhat constrained DUC environment to real driving, a number of issues must be addressed, pertaining to the ‘scaling-up’ of the system; cars will have to go faster and deal with higher traffic and obstruction (red lights, pedestrians, etc.) densities.

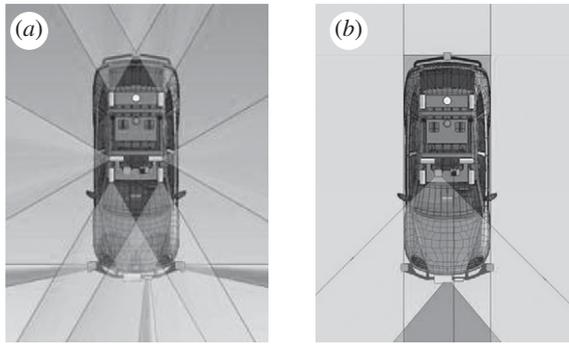


Figure 7. Sting sensor coverage producing an effective sensing range of around 80 m, resulting in a maximum speed of 50 km h^{-1} . (a) Six Prosilica GC 650 colour gigabit Ethernet cameras and three EVT-300 radars. (b) Five SICK LMS 291 LADARs and forward-looking Riegl LMS-Q120 LADAR.

One clear lesson from the DUC is that autonomous driving in traffic at 50 km h^{-1} (13.9 m s^{-1}) is significantly different from that at 10 km h^{-1} (2.8 m s^{-1}). This is partially owing to the drop in response time needed to support driving at those speeds. The response time T_{resp} is roughly an additive function of the time it takes (i) to sense the environment, (ii) to process the sensor data, (iii) to reach a control decision, (iv) to command the actuators, (v) for the actuators to respond and (vi) for the vehicle to respond (see Kelly & Stentz (1998) for a thorough discussion of this topic). Based on the maximal range Δ over which the sensors can detect other vehicles and obstacles (this range is typically obtained with radars or laser range sensors), a back-of-the-envelope calculation gives that the maximum speed supported by the system is proportional to Δ/T_{resp} . (The Georgia Tech vehicle's sensor footprint is shown in figure 7 together with the maximum effective range associated with the different sensing modalities.) All four teams represented by the authors performed this type of calculation, reaching a maximum top speed of between 40 and 50 km h^{-1} . (The maximum speed of the DUC was 30 mph (48 km h^{-1}).)

For example, Team Cornell's car showed that initial, fairly uncertain tracking started at about 150 m range ($\sigma \approx 10 \text{ m}$), followed by more accurate tracking at 60 m ($\sigma < 1 \text{ m}$). At 50 km h^{-1} , a car driving towards another car has approximately two seconds of accurate information to make decisions. At 100 km h^{-1} , a car has one second of accurate information. Clearly, either computation must be improved, or anticipation/reasoning algorithms must be able to execute extremely fast, or sensing must improve its range capability in order to increase the speeds of the vehicles.

As a consequence, the leap from 50 to 150 km h^{-1} is going to be highly non-trivial and, arguably, the hardware and software used in the DUC do not directly support this leap. For this to be possible, longer range sensing is required, together with a reduction in T_{resp} , e.g. by more dedicated perception and decision algorithms. As vehicle speeds increase, the dynamics of the vehicles will also become more of an issue, with slipping, skidding, and swerving all of a sudden playing a role, and full-scale dynamic friction models will be needed to understand these issues.

Another aspect of the ‘scaling-up’ phenomenon is the increased traffic density. As this density increases from the mild suburban driving in the DUC to truly urban or highway driving, a number of research issues must be resolved. These issues are quite different if the surrounding cars are travelling in the same direction (as on highways) or in opposite directions (as in urban driving), and we will treat these two cases as separate.

On highways, where the flow of the traffic goes in the same direction, new autonomous driving paradigms will become important beyond the ‘single-car’ paradigm, in which a safety zone is maintained around the vehicle. In fact, on highways, vehicles are executing provably unsafe manoeuvres all the time, based on the assumption that the surrounding vehicles are acting in a predictable and correct manner (like driving in the right direction and not arbitrarily slamming the brakes). For example, one can use platooning ideas in these environments, where the vehicles act together in tightly coupled formations. This way of organizing the driving would support a higher traffic density. Autonomous driving in platoons will have to rely on inter-vehicle communications in conjunction with sensing algorithms for its operation, and this line of thinking was at the centre of the California Partners for Advanced Transit and Highways (PATH) project, for example (Chang *et al.* 1991; Darbha 2002).

In urban environments, one can no longer make the assumption that the traffic flow is (more or less) uniform. Here, the higher traffic and obstruction densities require better sensing and tracking algorithms in that models of the intentions of surrounding vehicles are needed in order to be able to predict what they are doing. And to make matters worse, models are probably needed for how the surrounding vehicles will react to whatever action one takes as well, which is currently a largely unexplored area of research.

When driving through an urban environment, it is possible to encounter pedestrians, dogs, bicycles and other moving objects in addition to other cars. As such, it is important that the sensing algorithms faithfully manage to track and classify these obstructions since a collision would be highly undesirable. An explicit handling of the trade-off between performance and risk will become key for managing these types of environments in that there is a big difference between just ‘seeing’ an object and ‘knowing what it is up to’.

(f) Verification and validation

It is clear from the experiences at the DUC and the technical challenges described above that a key hurdle for autonomous systems will be the ability to verify that the system can operate safely and robustly. Safety is particularly important in any environment in which humans are present and particularly challenging environments in which humans control the agents that autonomous vehicles must interact with. The technologies trends and opportunities described in the previous sections must be integrated and tested in a manner that ensures proper operation.

Current methods for verifying that a specification is met typically involve substantial amounts of simulation and testing. Simulations can be very effective for checking large numbers of operating conditions in which the dynamics of the system and its environment can be captured by a model of appropriate complexity. Most simulation-based verification processes are augmented with

substantial systems testing. As the experiences of the DUC teams indicate, both of these techniques are essential for proper operation, but also fail to capture many important situations that can occur during actual operation. None of the authors' DUC teams applied formal reasoning about safety and performance of their system, in part owing to the tight time constraints but also owing to the lack of good tools that were compatible with the verification problems that arose. Some post-race formal analyses were performed (Wongpiromsarn & Murray 2008; Wongpiromsarn *et al.* 2009a), indicating the potential utility of such approaches.

Improvements in verification require new languages, formalisms and tools for specification, design and verification of autonomous systems. Existing work in computer science (algorithms, distributed computing systems, formal logic, model checkers) and system theory (control systems, signal processing, estimation theory, game theory) must be combined to handle dynamic operations better in uncertain, noisy and potentially adversarial environments. New techniques in hybrid systems (Frehse 2008), model checking (Lamport 2002; Holzmann 2003) and stability certificates (Parrilo 2003) provide promising directions for future research.

As an example of the types of techniques that can be brought to bear, consider the problem of navigation through an intersection when other cars are also present. The logic for handling this situation must properly react to combinations of other vehicles that are also at the intersection, when they arrive relative to your vehicle, and how to react if something unexpected happens. Typically, a fairly complex finite-state machine is used to keep track of the different modes of operation (queuing, waiting for intersection to clear, etc.), combined with sensor-based measurements of the actions and timing of other vehicles at the intersection. Once all of the code is written to implement this logic, it must be tested to verify proper operation. However, it is not possible to test for every possible combination of events at the intersection, especially when one starts to consider the complex dynamics of multiple vehicles stopping and starting as they react to the others.

A formal approach to this problem consists of generating a set of specifications that capture all of the possible motions at a high level as well as the proper (safe) operations of the vehicle. For example, we might require that we never proceed into an intersection if someone else has the right of way while at the same time requiring that we eventually traverse the intersection (to avoid a 'solution' in which we stop and never proceed). These specifications can be written precisely using linear temporal logic (Manna & Pnueli 1995), an extension of traditional logic that allows reasoning about sequences of actions. A specification written in linear temporal logic might take the form

$$(\phi_{\text{init}} \wedge \square\phi_{\text{env}}) \implies (\square\phi_{\text{s}} \wedge \diamond\phi_{\text{g}}), \quad (3.1)$$

where ϕ_{init} is a temporal logic formula that describes the possible initial conditions for the system, ϕ_{env} specifies the possible actions of the environment, ϕ_{s} describes the safety properties of the system and ϕ_{g} describes the progress goals. The temporal logic square symbol means that the formula following it should be true for all times and the diamond symbol means that the formula should eventually be true. Thus, equation (3.1) can be read as saying that if the vehicle starts in a given set of initial operating states and the environment satisfies some specification on its possible actions, the system should always maintain safe operation and eventually reach its goal.

A wide variety of tools exist for analysing such systems and verifying their correct behaviour. For purely discrete systems (no dynamics), two approaches to verifying temporal logic specifications are theorem proving and model checking. In theorem proving, one attempts to prove specifications by applying a sequence of logical steps that prove the desired statement is true. Tools such as PVS (Owre *et al.* 1998) allow this to be done in a semi-automated fashion. An alternative approach is model checking, in which one attempts to enumerate exhaustively all possible execution sequences and verify that the given temporal logic specification holds at each step. Tools such as SPIN (Holzmann 2003) and TLC (Lamport 2002) are able to efficiently enumerate all possible executions and can be used for quite large systems. The chief limitation in most existing approaches is the way in which dynamic processes are modelled and analysed, although symbolic model checkers are now available (see Pappas (2010) for a good overview) and we have applied these to several problems related to autonomous motion control (Braman *et al.* 2007*a,b*; Wongpiromsarn *et al.* 2009*a*).

Despite the progress in formal verification methods in the computer science, controls and hybrid systems communities, substantially more research is required to make these methods applicable to problems at the level of complexity of future autonomous systems. Research on ‘proof by construction techniques’ (e.g. Karaman *et al.* 2008; Kress-Gazit *et al.* 2009; Wongpiromsarn *et al.* 2009*b*) shows some promise, but still requires very simplified descriptions of the environment and system dynamics. Issues of robustness in formal verification, which are particularly important for reasoning about physical processes, have not yet received substantial attention.

4. Opportunities for autonomous vehicle technologies

While the eventual goal of fully integrated, autonomous vehicles is possible in the long run, there are several shorter term opportunities for applying the technologies described above in more limited situations. In this section we describe some of those opportunities.

(a) Vehicle safety systems

There are more than 40 000 deaths in automobile accidents each year in the United States alone—many owing to diminished capacity, such as drowsiness, inattention, incapacitation or intoxication. It is clear that advanced safety systems, warning systems, security, and new sensing and control systems could improve these numbers.

Sensing systems have already become standard on many cars. Cameras pointing out the back of the car turn on when a car is in reverse, thus giving the driver additional sensed information. Proximity radars coupled with warning bells at varying frequency announce to the driver that the car is coming closer to an obstacle.

While these sensing systems are becoming standard on many cars, just as anti-lock brakes did a number of years ago, a combination of sensing and active control for safety is making its way into luxury automobiles and trucks. Lane departure

systems have been developed, typically with vision, which monitors if a car moves outside of a lane. Range and speed proximity to cars in front of a vehicle are also being monitored using vision, radar or even lasers. Finally, new technology is being developed that attempts to monitor a driver's behaviour, including 'normal' versus 'tired'; key attributes include swerving in and out of the lane, and large changes in speed.

A key hurdle of these vehicle safety systems has been how to integrate autonomous override control, where the car actually performs an action in response to a sudden change in the environment. Automated parallel parking was recently introduced, which operates the vehicle in close proximity to others. But, important advancements are on the horizon for cases while the car is moving, including slowing a car to maintain safe distances while it is in cruise control (adaptive cruise control), actively moving the car back into lane upon departure, and applying the brakes when an accident is imminent.

It is envisioned that more and more safety components will be developed and integrated (Economist 2008) as sensing systems become cheaper, and active control systems become more reliable and verifiable. Fully autonomous cars, while still a way into the future, will be on the horizon soon after.

(b) Intelligent highways and roadway infrastructure

As described in the previous subsection, modern automobiles are increasingly equipped with electronics and software to modify the dynamics of the vehicle, including radar-based speed and spacing control systems, chassis control technologies for stability enhancement and improved suspension characteristics, active control of suspension and braking, and active restraint systems for safety. In addition, more sophisticated use of networking and communications devices will allow enhanced energy management between components and vehicle diagnostics with owner/dealer notification. These advances are now being built up to study increased interaction between vehicles and the roadway infrastructure, as automated highways and self-controlled vehicles move from the research laboratory into applications.

Several groups around the world have explored the use of distributed control for problems related to intelligent highway and transportation systems. These problems include increased interaction between individual vehicles to provide safer operations (e.g. collision warning and avoidance), as well as interaction between vehicles and the roadway infrastructure.

A representative example of this class of applications is the California PATH project (California Partners for Advanced Transit and Highways 2006). In 1997 the PATH project developed and demonstrated a system for allowing cars to be driven automatically down a freeway at close spacing, as shown in figure 8. By decreasing the spacing of cars, the density of traffic on a highway can be increased without requiring additional lanes. Additional work within the PATH project has looked at a variety of other systems for better managing traffic flow (California Partners for Advanced Transit and Highways 2006) and this has been followed by several other highway automation projects (Department of Transportation 2010; Intelligent Transportation Society of America 2010).



Figure 8. A platoon of cars driving down the San Diego freeway as part of the PATH project.

These applications are particularly challenging since they begin to link heterogeneous vehicles through communications systems that will experience varying bandwidths and latency (time delays) depending on the local environment. Providing safe, reliable and comfortable operations for such systems is a major challenge to which many of the problems described in the previous section are directly applicable.

(c) *Fully autonomous driving*

Fully autonomous driving concepts have been available for some time for tightly controlled environments, such as dockyards and loading docks (Durrant-Whyte 1996). For vehicles operating in more dynamic environments, automated driving is not available, but numerous driver aids, including adaptive cruise control anti-lock brakes, and skid control have been operational for several years. In addition, much more advanced driver assistance features have recently entered the market, such as heads-up displays, active parking, auto-braking to avoid collisions, and lane assist.

Given this trend, there are many future opportunities for fully autonomous driving, such as extending of mobility-on-demand systems to enable autonomous cars, trucks, and buses to travel automatically between locations of high demand and recharge stations, which would help address issues such as congestion, space and land use, pollution, and energy use. There is also an opportunity to develop systems to enable higher density traffic flow and reduced fuel consumption through platooning (e.g. California PATH) or faster throughput on urban settings through coordinated vehicle motion/signals and vehicle re-routing given a detection of traffic snarls. For example, Dickey *et al.* (2008) developed a model to predict the traffic flows from the surrounding intersections with an online signal optimization model to obtain the signal timing plan for the subsequent cycle. With automated vehicles, one could imagine an even more sophisticated system that obtains real-time information/routes from the cars to optimize the traffic flow using both the traffic signals and by providing route congestion feedback to the cars themselves.

5. Conclusions

In this paper, we have presented an overview of the current state of the art in autonomous vehicles and described some of the upcoming technical challenges and opportunities as we develop the next generation of robotic navigation systems. Our perspective has been motivated by our participation in the DUC, where numerous technologies had to be integrated to develop vehicles capable of driving in realistic urban environments. We believe that the chances for future research provide many opportunities for exciting work in robotics, controls, artificial intelligence and many other systems disciplines.

Much of the work described in this paper was supported by the US Defense Advanced Research Projects Agency (DARPA), under the DARPA Grand Challenge programme (Norm Whitaker, programme manager). R.M.M. would also like to acknowledge support from the US Air Force Office of Scientific Research (AFOSR) and the Boeing Corporation. J.P.H. would also like to acknowledge support from AFOSR (FA9550-08-1-0086). M.C. would like to acknowledge support from AFOSR (FA9550-05-1-0118).

References

- Bacha, A. *et al.* 2008 Odin: Team VictorTango's entry in the DARPA Urban Challenge. *J. Field Robot.* **25**, 467–492. (doi:10.1002/rob.20248)
- Bana, S. & Varaiya, P. 2001 Space division multiple access (SDMA) for robust ad hoc vehicle communication networks. In *Proc. Conf. on Intelligent Transportation Systems*, pp. 962–967. New York, NY: IEEE.
- Bohren, J. *et al.* 2008 Little Ben: the Ben Franklin racing team's entry in the 2007 DARPA urban challenge. *J. Field Robot.* **25**, 598–614. (doi:10.1002/rob.20260)
- Boyd, S., Ghosh, A., Prabhakar, B. & Shah, D. 2006 Randomized gossip algorithms. *IEEE Trans. Inform. Theory* **52**, 2508–2530. (doi:10.1109/TIT.2006.874516)
- Braman, J. M. B., Murray, R. M. & Ingham, M. D. 2007a Verification procedure for generalized goal-based control programs. In *AIAA Infotech@Aerospace Conf., Rohnert Park, CA, 7–10 May 2007*, AIAA-2007-3010.
- Braman, J. M. B., Murray, R. M. & Wagner, D. A. 2007b Safety verification of a fault tolerant reconfigurable autonomous goal-based robotic control system. In *Proc. Int. Conf. on Intelligent Robots and Systems*, pp. 859–864. New York, NY: IEEE.
- Brooks, A., Makarenko, A., Williams, S. & Durrant-Whyte, H. 2006 Parametric POMDPs for planning in continuous state spaces. *Robot. Auton. Syst.* **54**, 887–897. (doi:10.1016/j.robot.2006.05.007)
- Burdick, J. W., duToit, N., Howard, A., Looman, C., Ma, J., Murray, R. M. & Wongpiromsarn, T. 2007 Sensing, navigation and reasoning technologies for the DARPA Urban Challenge. DARPA Urban Challenge final report. See <http://gc.caltech.edu/media/papers/dgc07-final.pdf>.
- California Partners for Advanced Transit and Highways. 2006 See <http://www.path.berkeley.edu>.
- Chang, K. S. *et al.* 1991 Experimentation with a vehicle platoon control system. In *Proc. Conf. on Vehicle Navigation and Information Systems, Dearborn, MI, 20–23 October 1991*, pp. 1117–1124.
- Darbha, S. 2002 A note on string stability. *ASME J. Dyn. Syst. Meas. Control* **124**, 472–475. (doi:10.1115/1.1485099)
- Defense Advanced Research Projects Agency (DARPA). 2007 See <http://www.darpa.mil/grandchallenge>.
- Department of Transportation. 2010 National Transportation Library—intelligent transportation systems. See http://ntl.bts.gov/ref/biblio/its/its_references.html.
- Dickey, S. *et al.* 2008 Development of hardware in the loop simulation and paramics/VIS-PLUS integration. Technical report UCB-ITS-PRR-2008-29, California PATH research report. See <http://www.path.berkeley.edu/PATH/Publications/PDF/PRR/2008/PRR-2008-29.pdf>.

- Dickmanns, E. D. 2002 The development of machine vision for road vehicles in the last decade. In *Proc. IEEE Intelligent Vehicle Symp.*, pp. 268–281. New York, NY: IEEE.
- Durrant-Whyte, H. F. 1996 An autonomous guided vehicle for cargo handling applications. *Int. J. Robot. Res.* **15**, 407–440. (doi:10.1177/027836499601500501)
- Economist. 2008 Stopping in a hurry. *The Economist*, 11 December 2008.
- Economist. 2009 The connected car. *The Economist*, 4 June 2009.
- Fletcher, L. *et al.* 2008 The MIT-Cornell collision and why it happened. *J. Field Robot.* **25**, 775–807. (doi:10.1002/rob.20266)
- Frehse, G. 2008 PHAVer: algorithmic verification of hybrid systems past HyTech. *Int. J. Softw. Tools Technol. Transfer* **10**, 263–279. (doi:10.1007/s10009-007-0062-x)
- Gonzalez, J. P. & Stentz, A. 2007 Planning with uncertainty in position using high-resolution maps. In *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 1015–1022. New York, NY: IEEE.
- Hardy, J., Campbell, M., Miller, I. & Schimpf, B. 2008 Sensitivity analysis of an optimization-based trajectory planner for autonomous vehicles in urban environments. *Proc. SPIE* **7112**, 711211. (doi:10.1117/12.802599)
- Holzmann, G. J. 2003 *The SPIN model checker*. Reading, MA: Addison-Wesley.
- Huang, A., Olson, E. & Moore, D. 2009 Lightweight communications and marshalling for low-latency interprocess communication. Technical report MIT-CSAIL-TR-2009-041, MIT. See <http://dspace.mit.edu/bitstream/handle/1721.1/46708/MIT-CSAIL-TR-2009-041.pdf>.
- Iagnemma, K. & Buehler, M. (eds) 2006 Special issue on the DARPA grand challenge. *J. Field Robot.* (Special Issue) **23**, 461–835.
- Iagnemma, K., Buehler, M. & Singh, S. (eds) 2008 Special issue on the 2007 DARPA urban challenge. *J. Field Robot.* (Special Issue) **25**, 423–860.
- Intelligent Transportation Society of America. 2010 See <http://www.itsa.org>.
- Karaman, S., Sanfelice, R. G. & Frazzoli, E. 2008 Optimal control of mixed logical dynamical systems with linear temporal logic specifications. In *Proc. 47th IEEE Conf. on Decision and Control*, pp. 2117–2122. New York, NY: IEEE.
- Kelly, A. & Stentz, A. 1998 Rough terrain autonomous mobility—part 1: a theoretical analysis of requirements. *Auton. Robots* **5**, 129–161. (doi:10.1023/A:1008801421636)
- Kress-Gazit, H., Fainekos, G. & Pappas, G. 2009 Temporal-logic-based reactive mission and motion planning. *IEEE Trans. Robot.* **25**, 1370–1381. (doi:10.1109/TRO.2009.2030225)
- Kuwata, Y., Teo, J., Fiore, G., Karaman, S., Frazzoli, E. & How, J. P. 2009 Real-time motion planning with applications to autonomous urban driving. *IEEE Trans. Control Syst. Technol.* **17**, 1105–1118. (doi:10.1109/TCST.2008.2012116)
- Lamport, L. 2002 *Specifying systems*. Reading, MA: Addison-Wesley.
- Leonard, J. *et al.* 2008 A perception-driven autonomous urban vehicle. *J. Field Robot.* **25**, 727–774. (doi:10.1002/rob.20262)
- Manna, Z. & Pnueli, Z. 1995 *Temporal verification of reactive systems: safety*. Berlin, Germany: Springer.
- Miller, I. & Campbell, M. 2007 Rao-blackwellized particle filtering for mapping dynamic environments. In *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 3862–3869. New York, NY: IEEE.
- Miller, I. *et al.* 2008 Team Cornell's Skynet: robust perception and planning in an urban environment. *J. Field Robot.* **25**, 493–527. (doi:10.1002/rob.20253)
- Montemerlo, M. *et al.* 2008 Junior: the Stanford entry in the urban challenge. *J. Field Robot.* **25**, 569–597. (doi:10.1002/rob.20258)
- Olfati-Saber, R., Fax, J. A. & Murray, R. M. 2007 Consensus and cooperation in networked multi-agent systems. *Proc. IEEE* **95**, 215–233. (doi:10.1109/JPROC.2006.887293)
- Owre, S., Rushby, J., Shankar, N. & Stringer-Calvert, D. 1998 PVS: an experience report. In *Applied formal methods—FM-trends 98* (eds D. Hutter, W. Stephan, P. Traverso & M. Ullman). Lecture Notes in Computer Science, vol. 1641, pp. 338–345. Berlin, Germany: Springer.
- Pappas, G. 2010 Hybrid system tools. See <http://wiki.grasp.upenn.edu/hst>.
- Parrilo, P. A. 2003 Semidefinite programming relaxations for semialgebraic problems. *Math. Prog. Ser. B* **96**, 293–320.

- Pineau, J., Gordon, G. & Thrun, S. 2006 Anytime point-based approximations for large POMDPs. *J. Artif. Intell. Res.* **27**, 335–380.
- Rosencrantz, M., Gordon, G. & Thrun, S. 2003 Locating moving entities in dynamic indoor environments with teams of mobile robots. In *Proc. Int. Joint Conf. on Autonomous Agents and Multiagent Systems*, pp. 233–240. New York, NY: ACM.
- Santa, J., Gomez-Skarmeta, A. F. & Sanchez-Artigas, M. 2008 Architecture and evaluation of a unified V2V and V2I communication system based on cellular networks. *Comput. Commun.* **31**, 2850–2861. (doi:10.1016/j.comcom.2007.12.008)
- Sukthankar, R., Baluja, S. & Hancock, J. 1997 Evolving an intelligent vehicle for tactical reasoning in traffic. In *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 519–524. New York, NY: IEEE.
- Thorpe, C., Hebert, M., Kanade, T. & Shafer, S. 1988 Vision and navigation for the Carnegie-Mellon Navlab. *IEEE Trans. Pattern Anal. Mach. Intell.* **10**, 362–373. (doi:10.1109/34.3900)
- Urmson, C. *et al.* 2008 Autonomous driving in urban environments: Boss and the Urban Challenge. *J. Field Robot.* **25**, 425–466. (doi:10.1002/rob.20255)
- Wongpiromsarn, T. & Murray, R. M. 2008 Distributed mission and contingency management for the DARPA Urban Challenge. In *Proc. 2nd Int. Workshop on Intelligent Vehicle Control Systems*, pp. 19–29. Setubal, Portugal: INSTICC.
- Wongpiromsarn, T., Mitra, S., Murray, R. M. & Lamperski, A. 2009a Periodically controlled hybrid systems: verifying a controller for an autonomous vehicle. In *Hybrid systems: computation and control* (eds R. Majumdar & P. Tabuada). Lecture Notes in Computer Science, vol. 5469, pp. 396–410. Berlin, Germany: Springer.
- Wongpiromsarn, T., Topcu, U. & Murray, R. M. 2009b Receding horizon temporal logic planning for dynamical systems. In *Proc. 48th IEEE Conf. on Decision and Control*, pp. 5997–6004. New York, NY: IEEE.
- Wu, H., Fujimoto, R. & Riley, G. 2004 Analytical models for information propagation in vehicle-to-vehicle networks. In *Proc. 60th Vehicular Technology Conf.*, pp. 4548–4552. New York, NY: IEEE.