Toward Robotic Cars

By Sebastian Thrun

This article advocates self-driving, robotic technology for cars. Recent challenges organized by DARPA have induced a significant advance in technology for autopilots for cars; similar to those already used in aircraft and marine vessels. This article reviews this technology, and argues that enormous societal benefits can be reaped by deploying this emerging technology in the marketplace. It lays out a vision for deployment, and discusses some of the key remaining technology obstacles.

1. INTRODUCTION

Perhaps no invention has influenced the 20th century more than the automobile. Most of us use cars daily, as our primary mode of transportation. In fact, there are presently over 800 million cars on the road worldwide.[11] Car-related expenses constitute the second highest spending of the average American family, as 87% of the working population commute solely by car. As a result, cars consume approximately 34% of the Nation's energy.[9, 15]

Despite the importance of the automobile, insufficient innovation has occurred in past decades. Today, cars are *grossly inefficient* when it comes to basic resources, such as human health, energy, and human productivity. In the United States alone, 42,000 people die annually in nearly 6 million traffic accidents.[9] Traffic jams account for 3.7 billion wasted hours of human time and 2.3 billion wasted gallons of fuel.[5] And because of our strong emphasis on individual car ownership, cars are utilized less than 4% of their lifetime, wasting precious natural resources and space when not in use. The societal costs of our wasteful utilization of cars are truly staggering!

This article advocates robotic technology for making cars more efficient. The author conjectures that with suitable development efforts, robotic cars will critically enhance driver safety. They will reduce traffic congestion by reduced vehicle spacing and smoother driving. And they will increase resource utilization by enabling entirely new car sharing models.

The bulk of this article focuses on specific prototype vehicles. State of the art in robotic technology was showcased in a recent series of DARPA Challenges.[1, 2] The DARPA Grand Challenge required autonomous driving along desert trails. In 2005, Stanford's robot Stanley won this challenge. Two years later, Carnegie Mellon University's robot Boss won the Urban Challenge. This challenge required driverless vehicles to traverse 60 miles of urban streets.

This article reviews some of the critical technology behind these vehicles. It then lays out a technology road-map for building affordable and reliable robotic cars.

2. THE DARPA CHALLENGES

The DARPA Challenges were unique experiments by DARPA. To spur technology innovation, DARPA issues a series of competitions endowed with significant prize money. The original Grand Challenge, announced in 2003 and first executed in 2004, required driverless robotic cars to navigate a 142 miles-long course through the Mojave desert. DARPA offered 2$M for the fastest team that could navigate the course within a 10 h time limit. Even though the course followed well-defined desert trails, all participating robots failed within the first few miles. This outcome was often interpreted as evidence that the technology was not ready for prime time.[12]

DARPA repeated the Grand Challenge in 2005, albeit along a new route, and double the prize money. Just as in the previous year, the now-132 miles long route led through flats, dry lake beds, and along treacherous mountain passes. Out of 195 registered teams, DARPA selected 23 finalists. Four robots returned within the allotted time, with Stanford's Stanley claiming first place; see Figure 1a. The fact that four robots finished indicated a significant advance in technology within just 18 months.

In 2007, DARPA organized a new competition, the Urban Challenge. This competition took place in a mock city with paved roads. Eleven robots and about three dozen conventional vehicles navigated in a maze of city roads. When vehicles met, they had to obey California traffic rules. Carnegie Mellon University's robot "Boss," shown in Figure 1b, claimed first place, and Stanford's robot "Junior" came in second; see Figure 1c. The Urban Challenge added the challenge of other traffic. The problem solving capabilities required in the Urban Challenge were also more demanding than in the Grand Challenge, as robots had to make choices which way to travel.

Even though both competitions are far simpler than everyday driving, these challenges were milestones for the field of robotics. The capabilities demonstrated here went significantly beyond what was previously possible. Behind these advances were solid innovations in a number of core technologies that shall be discussed in turn.

3. TECHNOLOGY

3.1. Vehicle hardware

Stanley and Junior, shown in Figure 1a and c, are based on a 2004 Volkswagen Touareg R5, and a 2006 Volkswagen Passat Wagon, respectively. Both vehicles utilize custom interfaces to enable direct, electronic actuation of throttle, brakes, gear shifting, and steering. Vehicle data are communicated to Linux-based computer systems through CAN bus interfaces.

Figure 1. Autonomous robots in the DARPA Challenges: (a) Stanford's robot Stanley, winner of the Grand Challenge; (b) CMU's Boss, winner of the Urban Challenge (photograph courtesy of Carnegie Mellon University, Tartan Team); (c) Junior, runner-up in the Urban Challenge; and (d) UC Berkeley's Ghostrider motorcycle (photograph courtesy of Anthony Levandowski, Blue Team).

(a)

(b)

(c)

(d)

Nearly all relevant sensors are mounted on custom roof racks. In Stanley's case, five scanning laser range finders and a color camera are mounted pointing forward into the driving direction of the vehicle, for road recognition. Junior utilizes five different laser range finders; the primary sensor is a rotating laser that scans the environment at 64 scan lines and 10 Hz. Both vehicles are also equipped with radar sensors, for long range obstacle detection. The roof racks hold the antennae for the GPS inertial navigation system (INS). Computers are all mounted in the vehicle trunks. Stanley uses six Pentium M blades connected by Gigabit Ethernet, whereas Junior employs two Intel quad cores, all running Linux.

Clearly, both vehicles are conceptually similar. Perhaps the most important difference is that Stanley's environment sensors are pointed forward whereas Junior's sensors face in all directions around the vehicle. This design acknowledges the fact that Stanley only needs to sense the road ahead, whereas Junior must be aware of traffic in every direction.

3.2. Software architecture

Software is the primary contribution of both vehicles, as software is the *key* to robotic driving. Autonomous driving software generally factors into three main functional areas: perception, planning, and control. Perception addresses the problem of mapping sensor data into internal beliefs and predictions about the environment. Planning addresses the problem of making driving decisions. Control then actuates the steering wheel, throttle, brake, and other vehicle controls. Additional software modules interface to the vehicle and its sensors, and provide overarching services such as data logging and watchdog functionality.

Both vehicle's software architecture is modular. Modules run asynchronously and push data from sensors to actuators in a pipeline-fashion. Figure 2 depicts a diagram for Stanley's software; Junior's software is similar. The modularity of the software maximizes its flexibility in situations where the actual process times for data are unknown; it also minimizes the reaction time to new sensor data (which, in both vehicles, is about 300 ms).

Figure 2. Flowchart of Stanley software. The two dozen modules push data through a pipeline, comprising perception, planning, and control. Additional modules provide the IO to the vehicle, and global services such as logging.
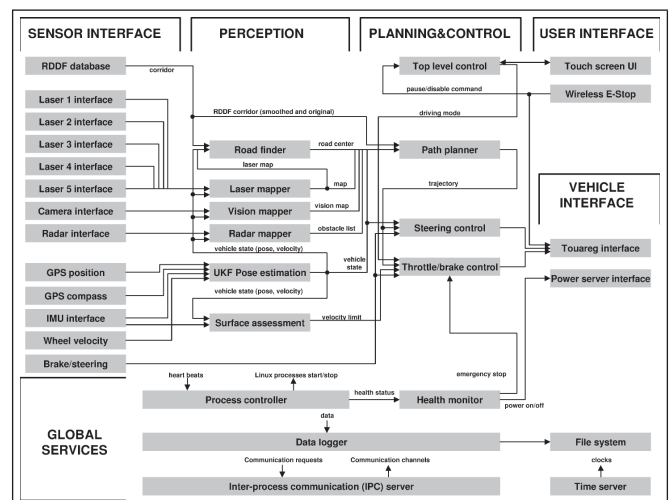
3.3. Sensor preprocessing

The "early" stage of perception requires data preprocessing and fusion. The most common form of fusion arises in the vehicle pose estimation, where "pose" comprises the vehicle coordinates, orientation (yaw, roll, pitch), and velocity. This is achieved via Kalman filters that integrate GPS measurements, wheel odometry, and inertial measurements.[13]

Further preprocessing takes place for the environment sensor data (laser, radar, camera images). Stanley integrates laser data over time into a 3-D point cloud, as illustrated in Figure 3. The point cloud is then analyzed for vertical obstacles, resulting in 2-D maps as shown in Figure 3. Because of the noise in sensor measurements, the actual test for the presence of a vertical obstacle is a probabilistic test.[14] This test computes the probability of the presence of an obstacle, considering potential pose measurement errors. When this probability exceeds a threshold, the map is marked "occupied."

A similar analysis takes place in Junior. Figure 4 illustrates a scan analysis, where adjacent scan lines are analyzed for obstacles as small as curbs.

Perhaps one of the most innovative elements of autonomous driving pertains to the fusion of multiple sensors. Stanley, in particular, is equipped with laser sensors whose range only extends to approximately 26 m. At this range, it is impossible to see obstacles in time to avoid them.

Adaptive vision addresses this problem.[3] Figure 5 depicts camera images segmented into drivable and undrivable terrain. This segmentation relies on the laser data. The adaptive vision software extracts a small drivable area right in front of the robot, using the laser obstacle map. This area is then used to train the computer vision system, to recognize similar color and texture distributions anywhere in the

Figure 3. Stanley integrates data from multiple lasers over time. The resulting point cloud analyzed for vertical obstacles, which are avoided.
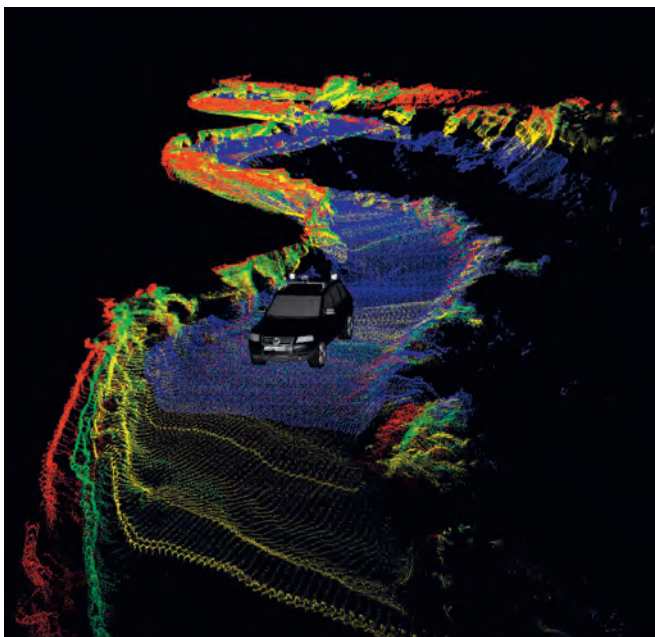
Figure 4. Junior analyzes 3-D scans acquired through laser range finer with 64 scan lines. Shown here is a single laser scan, along with the corresponding camera view of the vehicle.
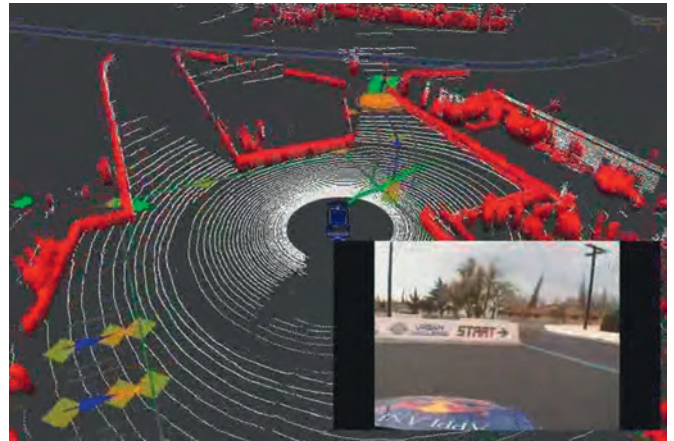
Figure 5. Camera image analysis is based on adaptive vision, which leveraged short-range laser data to train the system to recognize similar-looking terrain at greater distance.

image. The adaptation is performed ten times a second, so that the robot continuously adapt to the present terrain conditions. Adaptive vision enhances the obstacle detection range by up to 200 m, and it was essential in Stanley's ability to travel safely at speeds of up to almost 40 mph.

3.4. Localization

In both challenges, DARPA supplied contestants with maps of the environment. Figure 6 shows the Urban Challenge map. The maps contained detailed information about the drivable road area, plus data on speed limits and intersection handling.

Localization addresses the problem of establishing correspondence between the robot's present location, and the map. At first glance, the INS system may appear sufficient to do so; however, the typical INS estimation error can be a meter or more, which exceeds the acceptable error in most cases. Consequently, both robots relate features

**Figure 6. Course map provided by DARPA, here shown with our data processing tool.**

visible in the laser scans to map features, to further refine localization.

Stanley's localization only addresses the lateral location of the robot relative to the map. Figure 7 illustrates the analysis of the terrain for a discrete set of vertical offsets. Localization then adjusts the estimated INS pose estimates such that the center line of the road in the map aligns with the center of the drivable corridor. As a result, Stanley tends to stay centered on the road (unless, of course, the robot swerves to avoid an obstacle).

Junior's localization is essentially identical, but using infrared remission values of the laser in addition to range-based obstacle features. Infrared remission facilitates

Figure 7. Localization uses momentary sensor data to estimate the location of the robot relative to the map with centimeter precision. In Stanley, the localization only estimated the lateral location, as indicated by the lateral offset bars.
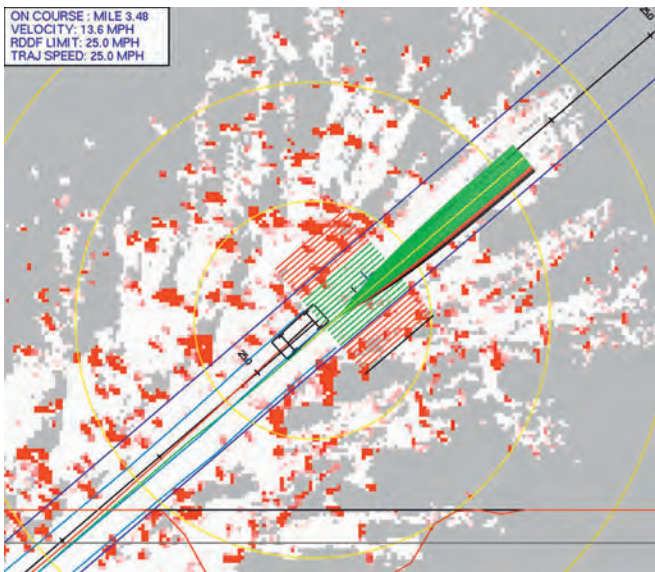
Figure 8. Lateral localization in Junior, based on infrared remission values acquired by the laser. The yellow graph depicts the posterior lateral position estimate.
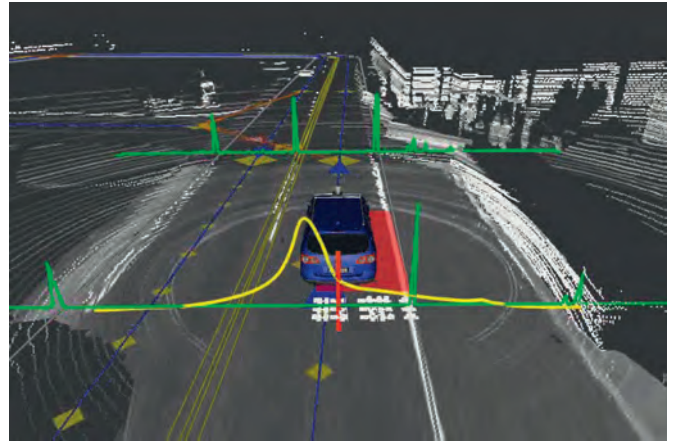
the detection of lane markings, which are not detectable with range. Figure 8 illustrates an infrared remission scan, superimposed with the localization results. The yellow curve in this figure represents the posterior distribution over the lateral offset to the map, as estimated by fusing INS and remission values. In this specific instance, the localizer reduces the GPS error from about a meter to a few centimeters.

3.5. Obstacle tracking

Roads are full of obstacles. Many are static, such as ruts and berms in the desert, or curbs and parked vehicles in an urban environment. To avoid such static obstacles, both vehicles build local occupancy grid maps[8] that maintain the location of static obstacles. Figure 9 shows examples of a maps built by both robots. Whereas Stanley distinguished only three types of terrain—drivable, occupied, and unexplored—Junior also categorizes the type of obstacles by height, which leads to an approximate distinction of curbs, cars, and tall trees, as illustrated in Figure 9b.

Equally relevant is the tracking of moving objects such as cars, which play a major role in urban driving. The key element of detecting moving objects is *temporal differencing*. If two subsequent laser scans mark a region as free in one scan, and occupied in another, then this joint observation constitutes a potential "witness" of a moving object. For the situation depicted in Figure 10a, Figure 10b illustrates such an analysis. Here scan points colored red or green correspond to such witnesses. The set of witnesses is then filtered (e.g., points outside the drivable map are removed) and moving objects are then tracked using particle filters. Figure 10c depicts an example result in which Junior finds and tracks four vehicles.

Further vehicle tracking is provided using radar sensors. To this end, Junior possesses three radar detectors, one pointing straight ahead, and two pointing to each side. The radars provide redundancy in moving object detection, and hence enhance the vehicle's reliability when merging into moving traffic; however, they are only used

Figure 9. (a) Mapping in Stanley, where terrain is classified as either drivable (white), obstacles (dark gray), or unknown (light gray). (b) Junior build more elaborate maps that distinguish curbs from vertical obstacles and overhanging trees.

(a)

(b)
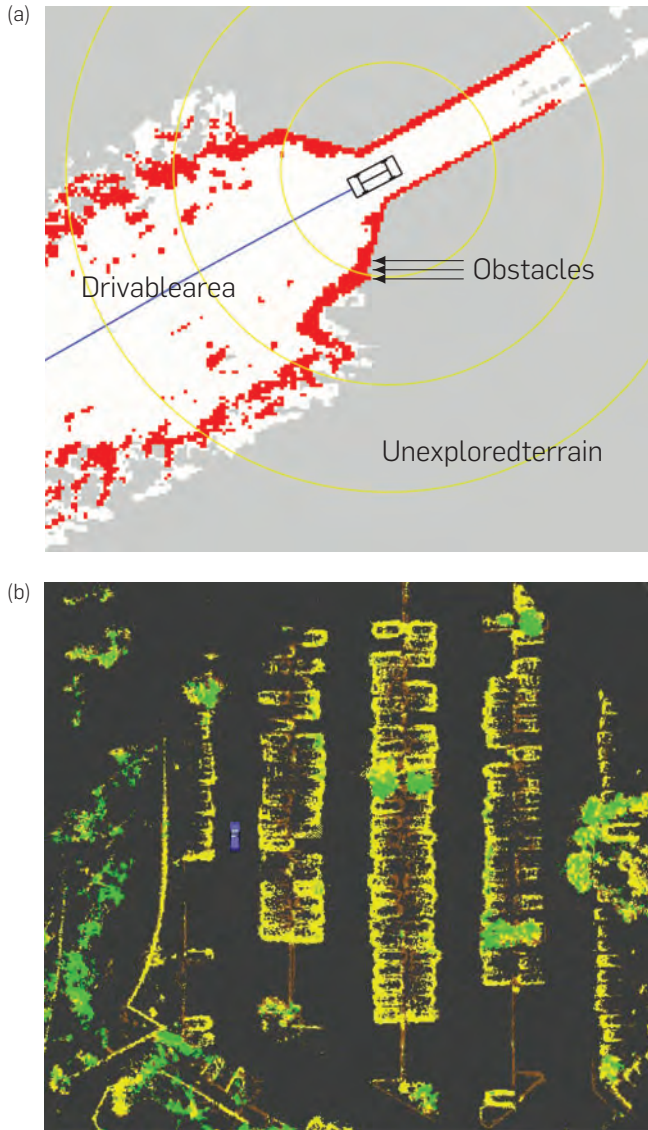
when the robot is standing still and attempting to merge into moving traffic.

3.6. Path planning

Driving decisions are made using path planning methods. Figure 11a illustrates a basic path planning technique, used by Stanley. This approach rolls out multiple trajectories to determine one that maximizes a plurality of criteria. These criteria minimize the risk of collision, but also favor the road centers over paths closer to the periphery. Search is performed along two dimensions: the amount of which the robot adjusts its trajectory laterally, and the speed at which this adjustment is carried out. Junior's basic path planning technique builds on the same idea, as illustrated in Figure 11b.

Figure 10. (a) Camera image of a scene in the urban challenge with oncoming traffic. (b) Scan differencing in this situation detects moving obstacles. (c) A particle filter tracks moving objects, as indicated by the boxes surrounding cars.

(a)

(b)

(c)
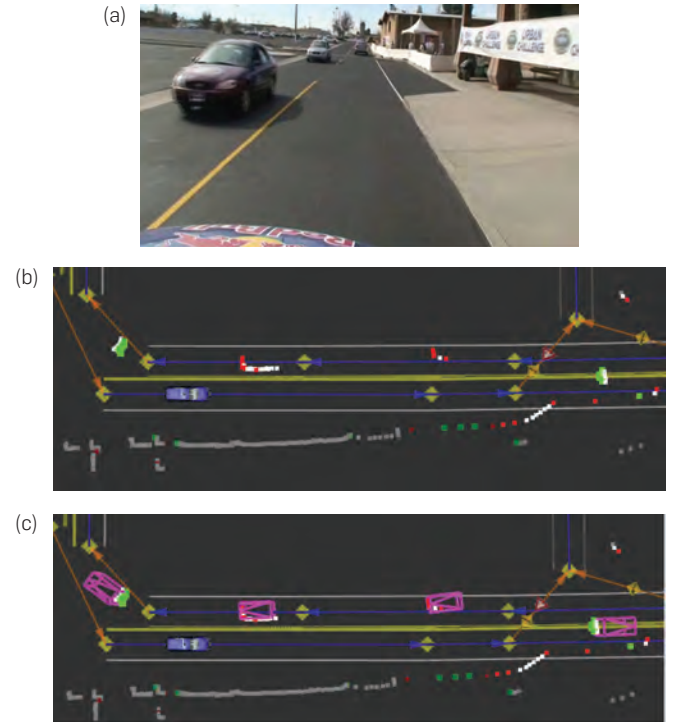
Planning in the Urban Challenge was substantially more demanding than in the Grand Challenge. In particular, the Urban Challenge required vehicles to choose their own path, and to navigate unstructured parking lots. For global path selection, Junior used a dynamic-programming-based global shortest path planner, which calculates the expected drive time to a goal location from any point in the environment. Hill climbing in this dynamic-programming function yields paths with the shortest expected travel time.

However, the momentary traffic situation may not permit driving the globally optimal path. Thus, Junior also considers local but discrete decisions, such as the lane change shown in Figure 11b, or the discrete turn decision illustrated in Figure 11c. In doing so, Junior minimizes its driving time in the context of the actual traffic situation. This approach permits Junior to react quickly and adequately to unforeseen situations, such as road blocks.

For "unstructured navigation" in parking lots, Junior uses a fast, modified version of the A* algorithm.[4] This algorithm searches shortest paths relative to the vehicle's map, using search trees like the one shown in Figure 12. The specific modification of conventional A* pertains to the fact that robot states are continuous, and consequentially, conventional A* is not guaranteed to find realizable paths. However, by caching continuous waypoints in search nodes expanded by A*, one can guarantee that any path found is indeed realizable.

A* planning usually requires less than a second, and is performed on the fly, as Junior maps its environment.

Figure 11. (a) Stanley rolls out potential paths to avoid collisions with obstacles. (b) Junior does the same, but also considers discrete choices such as lane changes. (c) Complex set of potential paths in a multi-intersection situation.
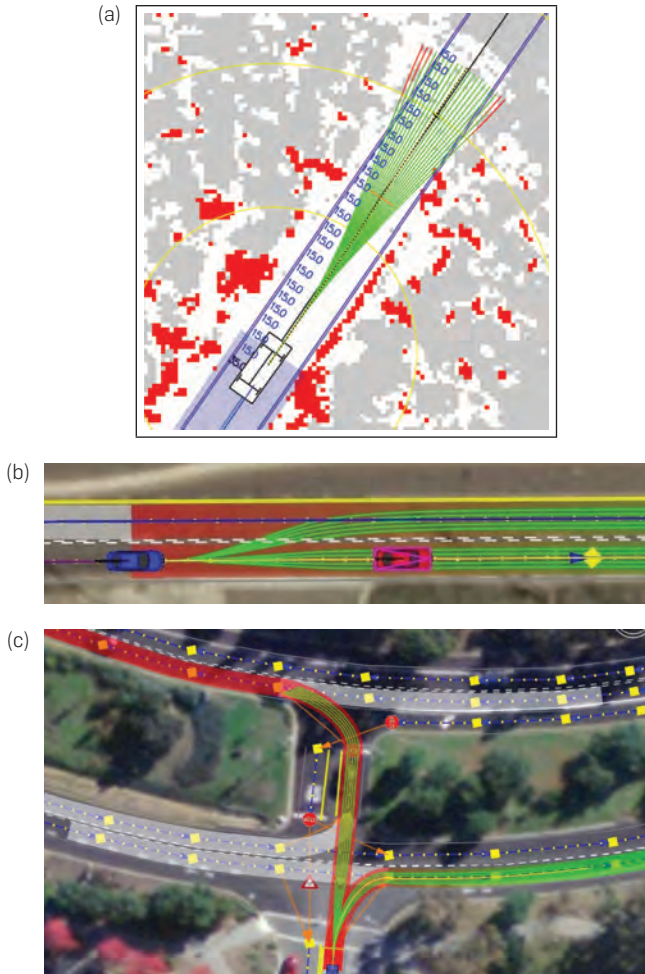

(a)

(b)

(c)

Figure 13 shows the application of this A* planner at a road blockage, where it generates a five-point U-turn.

3.7. Behaviors

Junior employs a behavioral module, which minimizes the risk of getting stuck in unpredictable environments. This module is implemented as a finite state machine, which controls the behavioral mode of the robot; see Figure 14. Under normal driving situations, driving behavior is governed by the appropriate path planner. However, when an impasse occurs, time-out mechanisms trigger to gradually permit increasingly unconstrained driving. Figure 15 illustrates this transition for a simulated traffic jam, where two other vehicles permanently block an intersection. After a time-out period, Junior invokes its unstructured A* path planner to find an unconstrained admissible route to its destination. The ability to gradually relax constraint in the driving process is essential for Junior's ability to succeed in situations as unpredictable as the Urban Challenge.

Figure 12. Junior finds paths to any target within milliseconds using a modified version of A*. Shown here are a search tree and the resulting path.

Figure 13. A* generate a U-turn maneuver for Junior, in response to a blocked road.
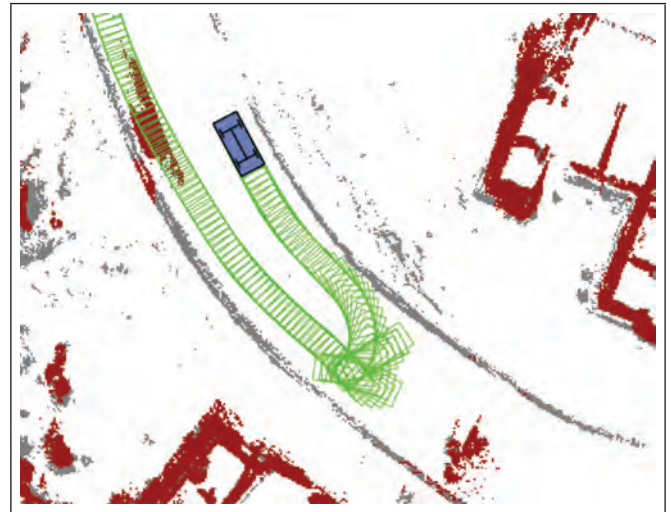

3.8. Control

The final software component realizes control of the vehicle itself, its throttle, brake, gear shifter, and steering wheel. In the actual race, steering and vehicle velocity were controlled using multiple PID controllers. Simply speaking, steering was adjusted so as to minimize any drift while pointing the front tires along the desired path. Gas and throttle were set so as to not exceed a maximum safe speed, calculated from path curvature, speed limits, and other obstacles (static and moving).

Figure 16 shows a backward side-sliding control maneuver, where the controller simultaneously controls for steering and vehicle speed. This LQR controller uses multiple control modes and vehicle models to transition

Figure 14. Junior's behavior is governed by a finite state machine, which provides for the possibility that common traffic rules may leave a robot without a legal option as how to proceed. When that happens, the robot will eventually invoke its general-purpose path planner to find a solution, regardless of traffic rules.
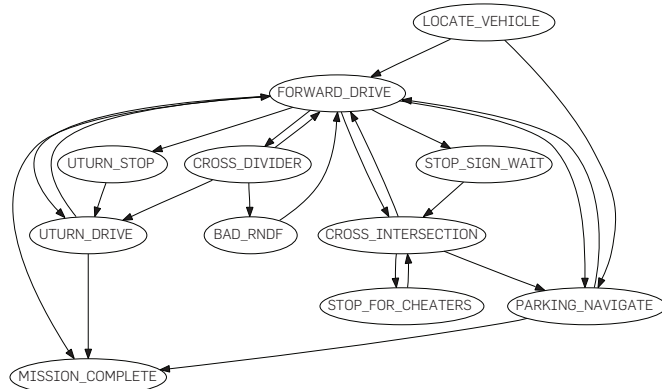
Figure 15. Resolving unexpected problems by invoking a general-purpose path planner after a wait period when facing a traffic jam. The detour is shown in (b).

(a)   (b)

Figure 16. Four snapshots of the car sliding backward into a parking spot.
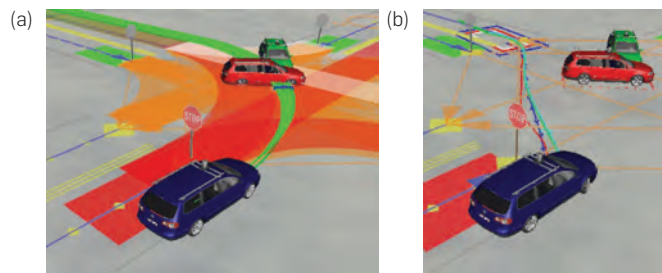
from conventional driving with full tracking, and sideward sliding.[6] This experiment illustrates the capabilities of Junior's present-day low level controls. Clearly, no sideways sliding was required (or desirable) in the DARPA challenges.

4. THE FUTURE

The results of the DARPA Challenges were limited in many ways, yet they point at a future where cars will be safer and a whole lot more convenient.

Future versions of the DARPA Challenge technology may be leveraged into new generation of cars that can "take over" during our daily commutes. Initially, such a take-over may only occur on highways, as limited-access highways are by far the easiest environments for robotic driving. Later, robotic cars may provide door-to-door chauffeur services, very much like a taxi cab without the driver. In doing so, we might exploit some of the advantages of robotic technology over human driving. For example, it seems technically feasible to operate robotic cars at distances of less than 10 m apart at highway speeds. This could double the throughput of highways relative to today, and also decrease energy consumption.[7] For the customer, such a technology would free

up significant time—in many cases more than an hour per working day.[1]

Robotic technology may also be leveraged to move unoccupied cars. At airports, rental cars may pick up their customers on the curbside, so no more waiting at the rental car counter. But the real potential lies in car sharing. As mentioned in the introduction to this article, cars are only utilized 4% of their lifetime. What if we could, on the click of a button, order a rental car straight to us. And once at our destination, we wasted no time looking for a parking; instead we just let the car drive away to pick up its next customer. Such a vision could drastically reduce the number of cars needed, and also free up important other resources, such as space consumed by parked cars. Perhaps in the future, most of us share cars, enabled through robotic technology.

All these visions require further advances in the technologies discussed. It seems pretty obvious that the main elements for autonomous driving are in place: perception, planning, and control. But there exists a range of challenges that present-day technology fails to fully address.

- One key challenge comes from "low-frequency" change. New roads are built; existing roads are repurposed; lanes are moved; construction zones may block lanes and alter the traffic flow. Any self-driving car needs to react adequately to such rare events. None of the DARPA challenges addressed these issues.
- Further, any robotic car must equal or surpass human reliability. All components of these cars—the sensors, computers, actuators, operating systems, and software—must become orders of magnitude more reliable to meet these goals. The DARPA Challenges established new milestones for robotic autonomy and reliability, but these systems are still far too unreliable for practical use.
- Finally, people need to feel comfortable in a robotic car. There is an urgent need to develop user interfaces and modes of control that make people feel comfortable with this new concept. Research is needed on the type information provided to the human driver and on modes to integrate human and robotic control. There

may be situations that even robotic technology will be unable to handle, which raises questions on how to best leverage human intelligence and driving skills should such situations occur.

Still, the only way to turn this new type of transportation into reality is to invest massively into the vision of smart, robotic cars. The benefits to society will be enormous. We need to overcome the old belief that only people can drive cars, and embrace new modes of transportation that utilize the twenty-first century technology. When this happens, we will free up significant resources that are presently wasted in the inefficiency of today's car-based society.

Acknowledgments

References

1. Buehler, M., Iagnemma, K., Singh, S. (eds.). *The 2005 DARPA Grand Challenge: The Great Robot Race.* Springer, Berlin, 2006.
2. Buehler, M., Iagnemma, K., Singh, S. (eds.). *The 2005 DARPA Urban Challenge: Autonomous Vehicles in City Traffic.* Springer, Berlin, 2009.
3. Dahlkamp, H., Kaehler, A., Stavens, D., Thrun, S., Bradski, G. Self-supervised monocular road detection in desert terrain. In *Proceedings of the Robotics Science and Systems Conference.* Sukhatme, G., Schaal, S., Burgard, W., Fox, D. (eds.) (Philadelphia, PA, 2006).
4. Dolgov, D., Montemerlo, M., Thrun, S. Path planning for autonomous driving in unknown environments. In *Proceedings of the International Symposium on Experimental Robotics (ISER)* (Athens, Greece, 2008). Springer Tracts in Advanced Robotics (STAR).
5. The extent of and outlook for congestion: Ministerial meeting on mitigating congestion, 2007. http://www.internationaltransportforum.org/sofia/sofiadocs.html.
6. Kolter, G., Plagemann, C., Jackson, D., Ng, A., Thrun, S. Hybrid optimal open-loop probabilistic conrol, with application to extreme autonomous driving. Submitted for publication, 2009.
7. Michaelian, M., Browand, F. Field experiments demonstrate fuel savings for close-following. Technical Report PATH UCB-ITS-PRR-2000-14, University of California at Berkeley, 2000.
8. Moravec, H.P. Sensor fusion in certainty grids for mobile robots. *AI Mag. 9,* 2 (1988), 61–74.
9. *National Transportation Statistics.* Bureau of Transportation Statistics, Department of Transportation, 2006.
10. *Omnibus Household Survey Information American Commuting Time.* Bureau of Transportation Statistics, Department of Transportation, 2006.
11. Passenger cars, 2006. http://www.sasi.group.shef.ac.uk/worldmapper/display.php?selected=31.
12. Singer, P.W. *Wired for War.* Penguin Press, 2009.
13. Thrun, S., Burgard, W., Fox, D. *Probabilistic Robotics.* MIT Press, Cambridge, MA, 2005.
14. Thrun, S., Montemerlo, M., Aron, A. Probabilistic terrain analysis for high-speed desert driving. In *Proceedings of the Robotics Science and Systems Conference.* Sukhatme, G., Schaal, S., Burgard, W., Fox, D. (eds.) (Philadelphia, PA, 2006).
15. *Transportation Energy Data Book (Edition 27).* US Department of Energy, 2008. http://cta.ornl.gov/data/index.shtml.

Sebastian Thrun (thrun@stanford.edu), Computer Science Department, Stanford University, Stanford, CA.