

PROUD—Public ROad Urban Driverless test: architecture and results

Alberto Broggi, Pietro Cerri, Stefano Debattisti, Maria Chiara Laghi,
Paolo Medici, Matteo Panciroli, Antonio Prioletti

Abstract—The presence of autonomous vehicles on public roads is becoming a reality. In the last 10 years, autonomous prototypes have been confined in controlled or isolated environments, but new traffic regulations for testing and direct automotive companies interests are moving autonomous vehicles tests on real roads. This paper presents a test on public urban roads and freeways that was held in Parma on July 12, 2013. This was the first test in open public urban roads with nobody behind the steering wheel: the vehicle had to cope with roundabouts, junctions, pedestrian crossings, freeway junctions, traffic lights, and regular traffic. The vehicle setup, the software architecture, and the route are here presented together with some results and possible future improvements.

I. INTRODUCTION

The development of autonomous vehicles has been speeding up in the last few years, thanks to the increasing interest of automotive companies. This paper presents a test of autonomous driving in public urban roads and freeways in Parma, including two-way narrow rural roads, pedestrian crossings, traffic lights, roundabouts and freeway junctions; the test was named PROUD 2013 (Public ROad Urban Driverless test). The test was a demonstration of the capabilities of our autonomous vehicle to drive in real traffic. As mentioned, different road environments were managed: although our BRAiVE vehicle prototype was not yet ready to manage all possible situations in real roads, the test demonstrated that the most common infrastructures can be correctly managed.

The new regulations for autonomous vehicles are allowing to test prototypes on real roads in some states. Special authorizations to hold this test were granted by Italian agencies, as Italy does not have a specific regulation for autonomous vehicles. One of the most interesting consequence of these new laws is the opportunity to evaluate how driving in real traffic can be managed by autonomous vehicles. Driving in well-structured environments, and with other road users moving with safe, correct, and predictable attitude, is an ideal situation, but the integration with normal traffic is a mandatory test for autonomous vehicles to reach the market.

The development of an autonomous vehicle architecture is a complex problem, differently approached by research groups. During the DARPA challenges (in 2004, 2005, and 2007) various architectures were developed. The TerraMax team [1] divided the architecture in two parts: the system services and the autonomous services. The autonomous services

were composed by three main parts: the World Perception Server (WPS), the Autonomous Vehicle Manager (AVM), and the Autonomous Vehicle Driver (AVD). The WPS was in charge of gathering information by perception servers and fuse them to build a perception map; this map was used by the AVM to manage the behavior and plan the vehicle path. The AVD communicated to the drive-by-wire, and it converted path into low level commands. Stanford's Junior [2] was based on a different architecture: Global Services, Sensor Interface, Perception, Navigation, User Interface and Vehicle Interface were the layers in which the system was divided. The most interesting parts for our purpose is the Perception layer. Three modules composed this layer, the UKF Pose Estimation, the Static and Dynamic Obstacle Tracker and the RNDF Localization & Repair: using this three modules it is possible to reach a precise positioning on the map and perceive the obstacles.

The BRAiVE architecture uses a WPS, similar to the one proposed by the TerraMax team, and a module similar to the UKF Pose Estimation proposed by Stanford university. The main architecture, presented in section IV, is a new concept developed starting from [3].

II. OUR APPROACH

The underlying idea of this test was to engage a new challenge, i.e. autonomous driving in open public roads. In the past years different challenges were held, with the aim of testing autonomous prototypes in safe and controlled environments. Presently, autonomous vehicles tests can be described according to different characteristics:

- Map: mapped/unmapped
- Scenario: familiar/unknown
- Environment: on-road/off-road
- Obstacles: static/dynamic and predictable/dynamic and unpredictable
- Speed: slow/fast

According to this taxonomy, driving in a mapped and familiar scenario is the first challenge that has to be accomplished: complexity grows if one of these constraints is not fulfilled. The off-road driving is considered more complex than driving in a structured environment. The other critical aspect is represented by obstacles: static obstacles, that can also be mapped, are quite easy to manage, but robust perception algorithms are mandatory in case of dynamic obstacles. If obstacles behavior can not be predicted, the perception has to be very reactive and the path planner must also consider this issue. The complexity grows with the speed, as well: 40/50

*This project was carried out in the frame of the ERC OFAV Grant received by Prof. Alberto Broggi.

Authors are with VisLab – Dipartimento di Ingegneria dell'Informazione Università degli Studi di Parma, ITALY <http://www.vislab.it> {broggi,cerri,deba,laghi,medici,panci,prio}@vislab.it

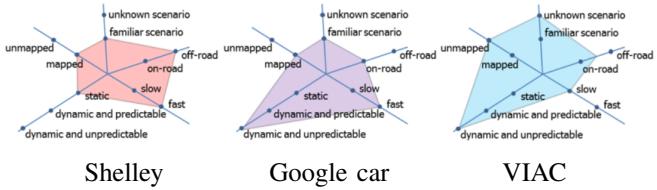


Fig. 1. Taxonomy of different autonomous vehicles tests.

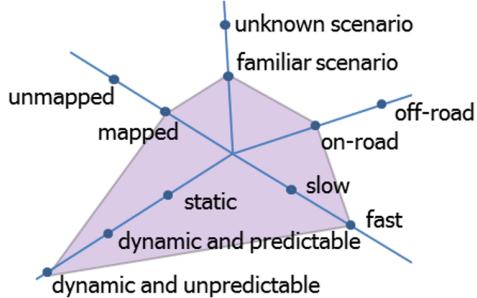


Fig. 2. PROUD challenge characteristics.

km/h speeds are considered slow, while higher velocities are considered fast.

Some examples of autonomous challenges are presented in figure 1. Shelley, the autonomous vehicle by Stanford University [4] was tested on a well known and well mapped environment at very high speeds, managing static obstacles. The Google car [5] can drive fast on familiar and well mapped road, managing any kind of obstacle. During VIAC (VisLab Intercontinental Autonomous Challenge) [6], the vehicles were able to drive in unmapped and unknown scenarios, managing any kind of obstacles, at low speeds.

Other challenges [7], [8], [9] were based on the assumption of dynamic obstacles moving with safe and predictable attitude.

The idea of PROUD is to move the VIAC experience to a different level: moving in a mapped and familiar scenario, it is possible to drive faster and to trigger the correct driving mode. Figure 2 describes the PROUD challenge using the presented taxonomy.

III. THE BRAIVE VEHICLE

BRAIVE has already been presented in other papers like [10]. Only a few modifications were made to enhance its performance. The Hokuyo UTM-30LX were replaced with the new UTM-30LX-EW models that are rated IP67 according to Ingress Protection Rating. Another important enhancement of this device is the multi-echo support, that is extremely useful in case of bad weather or dust. Both the Hokuyo devices are 270° single layer laser scanners, with a detection range that can reach 30 meters. The GNSS device was substituted with a Topcon AGI3, installed in the boot. This device is connected to an additional antenna mounted on the vehicle roof, and with a GSM antenna mounted on the surface of the boot. This antenna is used to receive RTK correction provided in real-time by GEOTOP, with the aim of reaching positioning error

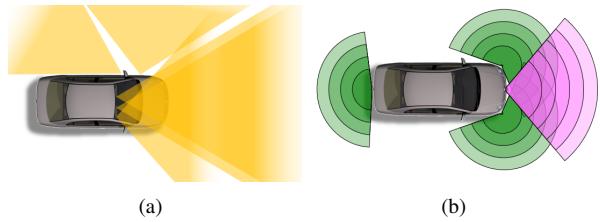


Fig. 3. Sensors actually used during the PROUD test. (a) Cameras and (b) lasers coverage.

lower than 5 centimeters in ideal conditions. The IBEO Lux output trigger signal is used to synchronize all the cameras.

Four computers are connected to the sensors trying to equally split the processing load. One PC is completely devoted to vehicle control: high, medium, and low level controls plus World Perception Server run on it. This computer is also connected via CAN Bus to the dSpace MicroAutoBox, in charge of communication to the vehicle Electronic Central Units. The other 3 computers are used to run the perception software. The first PC is connected to the stereo camera framing the area in front of the vehicle, and to the IBEO 4-layer laser scanner. This PC is used to detect frontal obstacles and lane markings. The second PC is connected to the two single layer laser scanners mounted in front bumper corners, and to the side looking cameras mounted over the front wheels. This PC is used to manage all lateral obstacles, especially when approaching roundabouts and junctions. The third PC is connected to the single-layer laser scanner mounted in the rear bumper, to the cameras mounted in the external rearview mirrors, facing backwards, and to a frontal camera. This PC is used to detect rear obstacles, obstacles approaching on the adjacent lanes, road markings (pedestrian crossings, stop lines), and traffic lights.

IV. SYSTEM ARCHITECTURE

For the PROUD test we built a specific architecture, aiming at creating a fluid, modular, and scalable system. The created architecture features a total separation between the perception, planning, and control layers; it provides a multilevel sensor fusion and works with open source data, especially regarding the navigation maps. This architecture is the final result of the Open intelligent systems for Future Autonomous Vehicles (OFAV) project.

The modularity and scalability target has been reached developing each component as an independent service, which communicates with the other components through a message passing service, called *VisLab Bus* (VLBus). VLBus is an abstraction that models the inter-process communication as a data bus: there can be different data buses, each with lines which represent different communication topics. VLBus uses shared memory to exchange information between applications and threads on the same PC and for network communication in different physical machines, in a transparent and efficient way.

The architecture is composed by three main layers, called perception, planning, and control layers. The perception

and planning layers have been separated by a component, called *World Perception Server* (WPS), which performs data fusion and acts as data provider to the planning layer. The WPS, which provides two of the main characteristics of the architecture (layers separation and multilevel data fusion), will be described more in detail later in this section together with some of the perception applications. All the perception applications running during PROUD run in VisLab's framework called *GOLD* [11], while the WPS is a standalone application, like all the modules of the planning layer. Due to the great complexity of the automotive scenario, especially regarding the planning and decision making phase, a hierarchical approach is adopted, splitting the planning layer in three sub-layers: Navigation layer, Maneuver layer, and Control layer [12]. The Navigation and Maneuver layers will be explained in section IV-A, while the Control layer will be explained in section IV-B.

A. Navigation and Maneuver

The aim of the upper layer (Navigation) is to generate a route, i.e. a set of roads and junctions, that links the actual position of the vehicle with the destination specified by the user. In this planner we use maps, only as cartographic source, coming from the "Open Street Map Project" [13]; we selected this project in order to fulfill the aforementioned constraint of openness of the used data. Starting from these maps, we created a modified compressed map, called *Road Junction Map*, which is represented by a graph in which each edge is a road segment and each node represents a fork between two or more roads. The *Road Junction Map*, compared to the corresponding OSM map has the main advantage of having approximately 40% less nodes, and therefore allows a much faster path planning. Route planning is based on a heuristic planner, derived from A* [14], where the cost function $c(x) = d(x_i, x_j)v(x_{i,j}) + \delta$ is based on the travel time of an edge plus a constant δ that has the aim to avoid unwanted path changes. The heuristic function of the planner is $h(x) = D(x, x_f)k_v$, which represents the theoretical travel time using the euclidean distance and an average speed k_v .

The Maneuver Layer is composed by a set of different maneuvers, including for example a lane keeping maneuver and a vehicle following maneuver, grouped in high-level sets called *Driving Modes*. The objective of the Maneuver layer is to produce a traversability map for the Control layer path planner, choosing the right maneuver from the maneuver database considering both road and traffic situations. Maneuvers, Driving Modes, and the relationships between them are described in more detail in [15]. Each maneuver represents an implementation of the base maneuver structure, which is mainly composed by a list of pre-conditions, a list of post-conditions, a list of exit-conditions, and a list of gates.

The gate concept is the basic idea of this level of planning: a gate integrates information about speed, position and orientation that the vehicle must have in the future trajectory without directly planning an entire trajectory [15]. Basically, a gate is a kind of checkpoint planned in the future trajectory

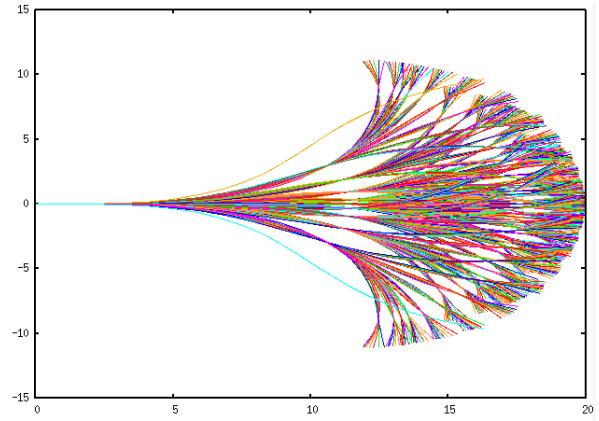


Fig. 4. Example of the precomputed trajectories.

through which the vehicle must pass. Starting from the gates list it is possible to create the traversability map needed by the Control layer path planner. A key aspect of the Maneuver layer is that it does not deal with obstacles and with their trajectory, because they are considered only in the Control layer. As said earlier, maneuvers are grouped in sets, called *driving modes* (DM); therefore each DM is composed by many different maneuvers that describe the various ways in which the target can be reached. In every DM each maneuver has a priority value and the trajectory planner of the Control Layer will select and execute the maneuver with the highest priority.

B. Path Planner

The path planner is based on several concepts developed for VIAC [16]. Since both steering and acceleration are controlled by the planner, this system includes obstacle avoidance and collision mitigation functions.

The WPS provides the list of static and dynamic obstacles while the Maneuver Level provides the traversability area. Using only static obstacles and the traversability area, a potential map is generated where a low potential is a good path to follow and a high potential is an area to avoid. Feasible trajectories are generated in space coordinates and evaluated on this map (see figure 4) considering the whole vehicle volume. The lower potential trajectory is considered the best one and reported to the planner. The equation of vehicle motion is

$$\begin{aligned}\dot{x} &= \cos \theta(s) \\ \dot{y} &= \sin \theta(s) \\ \dot{\theta} &= \kappa(s)\end{aligned}\quad (1)$$

where (x, y, θ) are position and orientation of vehicle and the curvature $\kappa(s)$ is a specific function. To generate trajectories with high smoothness, $\kappa(s)$ was chosen belonging to the class of quadratic splines.

In order to improve performance, initially only a rough subset of offline precomputed curves is evaluated. Up to 6 GB of RAM are used to store the precomputed trajectories, subdivided in 4 classes, according to current speed (0-5 m/s,

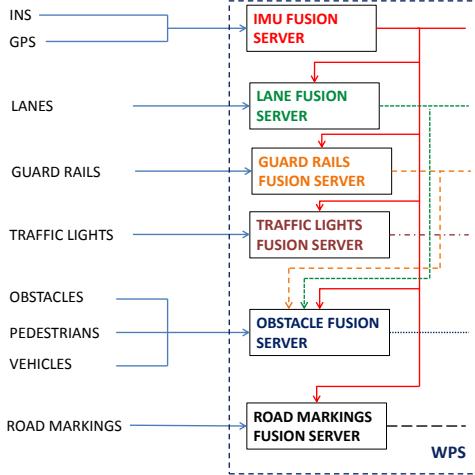


Fig. 5. Scheme of WPS fusion servers

5-10 m/s, 10-15 m/s, over 15 m/s) and indexed by initial vehicle curvature. Each class of trajectories is characterized by the curvature limits, typical of the associated speed. The best of the precomputed trajectories is used as initial estimation for a non linear optimization stage forcing continuity with the previous manouver.

The computed trajectory is a collision free trajectory based on static obstacles and it is represented in space-domain. A collision map in time-space coordinates is produced using this trajectory, the maximum lateral acceleration limit, information about maximum speed per point and the motion of dynamic obstacles. Using this map, a speed profile is computed in order to mitigate collision with dynamic obstacles and with the goal of being smooth and comfortable.

Finally, the trajectory and speed profile are provided to the low level control, where a Model Predictive Control (MPC) is used to convert space-domain $\kappa(s)$ to time-domain steering wheel set point $\delta(t)$ according to kinematic vehicle model. To deal with different weather, road conditions, and vehicle set-up, vehicle parameters are evaluated online constantly comparing the predicted path to the performed one.

C. World Perception Server

In order to manage and fuse data from all the sensors, the World Perception Server (WPS) is used. The WPS is responsible for receiving data from the perception and creating a unique global perception map.

Examples of WPS output are shown in figure 6. It is possible to distinguish two types of users that communicate with the server: the suppliers (sensors) and the clients (planners). A two-way handshake is requested to register to the server.

1) *Sensor-Side*: During the registration phase, perception applications send information regarding the type of perception that is being provided, the presence of tracking information, sensor position and field-of-view. If the WPS supports this perception type, it will acknowledge with a positive response and with an ID of the channel where the application will provide its data. Otherwise, if it can

not manage this type of information, it will answer with a “refused” message or, if the WPS is not ready, it will provide a “try later” message. Figure 5 shows the internal structure of the WPS, with several fusion servers for the different perception types. Each server receives data periodically from all the sensors that have been registered for that perception task. If an application is not sending data for a certain period of time, it gets unregistered; if there are no more sensors to provide that kind of perception, the fusion of this data is removed and its clients notified.

2) *Planner-Side*: The planner initially sends a message to assess the capabilities owned by the server. The WPS response contains the supported capabilities and, for each a channel pair: one to exchange command/information messages with the perception server and one for the server to provide the requested data. Once registered, the planner can request the fused data to the WPS specifying the time, in the future, when data must be projected.

As shown in Fig. 5, the WPS can manage six types of perception: inertial, obstacles, lanes, barriers, traffic lights, and road markings. Each request is dispatched to the specific fusion server that deals with it and returns the response to the main server. A more detailed analysis of each single fusion server follows.

3) *Obstacles*: An obstacle consists of a list of its 2D points projected on the ground. They are divided into static and moving obstacles with a state certainty that grows with the increase of the time it remains in the same state. The obstacles bounding boxes are defined as the minimum box that contains all obstacle points; two obstacle, coming from different applications, are associated if the minimum distance between their bounding boxes is less than the maximum position uncertainty of the two obstacles. Obstacles coming from the same application are associated by the WPS only if the specific application does not support tracking. A Kalman filter has been used to track obstacles, using the centroid to compute their speed. In the first step, obstacles from different kinds of application are not fused, since using high level fusion, based on obstacles centroids, leads to very different centroids when the obstacles come from different kinds of sensor. This leads to have different Kalman filters for each application, and obstacles are merged after the Kalman tracking process. Then, a lane membership is assigned to each obstacle. Information from barrier fusion server is used to decide if an obstacle belongs to a barrier and must be considered as static, even if the obstacle fusion server classifies it as dynamic.

4) *Lanes*: Lane markings are represented by polylines. It is possible to distinguish two levels of fusion: low-level fusion and high-level fusion. In the low-level one, lane markings coming from the sensors are associated and merged. The association process consists in a segment-to-segment matching of the two polylines: the two polylines are reconstructed in order to have the same number of segments and each segment is compared with the closest one of the other lane markings; if the average distance is under a certain threshold the lane markings are merged. The high-level step

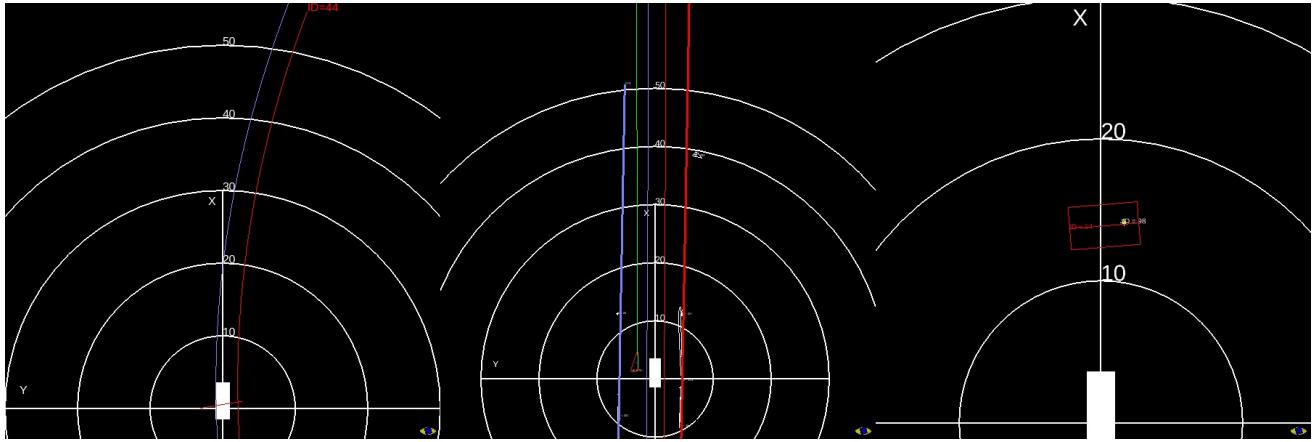


Fig. 6. WPS output examples. Left: an example of lane and stop line detection. Center: an example of barrier, lane and obstacle detection. Right: an example of zebra crossing and pedestrian detection.

outputs the left, right, and own lane. These are computed in three steps: first the medium points of the two lane markings that form the lane are evaluated, then a parabola is used to fit these points and, finally, a low-level filter is applied to remove noises spikes in the lane position in consecutive frames and stabilize the results for the navigation.

5) *Barriers*: Barriers fusion is very similar to lanes fusion since their internal representation is the same. The difference regards the high-level step : it outputs left and right barriers computed fitting a parabola on the polylines points.

6) *Traffic Lights*: A traffic light is represented by its position and a color corresponding to the turned on lamp. A simple association, based only on position, is used in the tracking stage, since it is not necessary to know the precise traffic light distance.

7) *Road Markings*: A road marking represents a pedestrian crossing, a stop line, or a highway junction, by a segment. A minimum segment-to-segment distance metric is used to associate two road markings of the same type.

8) *IMU*: The Inertial Measurement Unit provides two different types of information: precise relative motion of the vehicle and rough global position localization. 3 MEMS gyroscopes and accelerometers fused with precise wheels odometry providing relative vehicle motion; a GNSS with differential correction is used to supply a precise localization. When a condition of low satellites number visibility occurs, the inertial sensors provide a vehicle position estimate, clearly affected by drift. A Sigma Point Kalman Filter is used to merge all those data together.

V. TEST RESULTS

The test was conducted in Parma from the University campus to the town center through different scenarios such as urban, rural, and highway roads; a roadmap is shown in figure 7.

BRAIVE completed the whole track autonomously, with no human intervention, along a 13 km route divided into 0.7 km of rural roads, 1.7 km of urban environments and 10.6



Fig. 7. Track overview with the vehicle's path superimposed on the map.

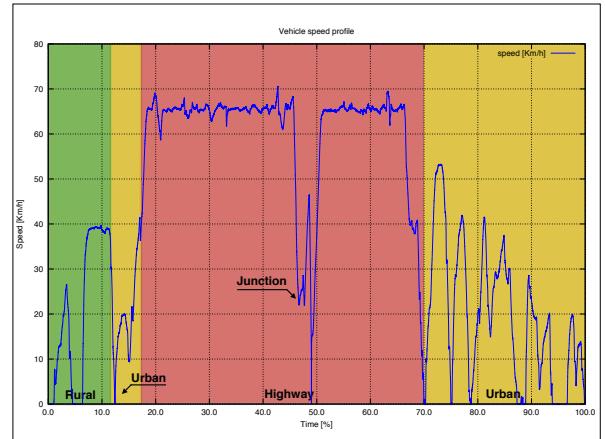
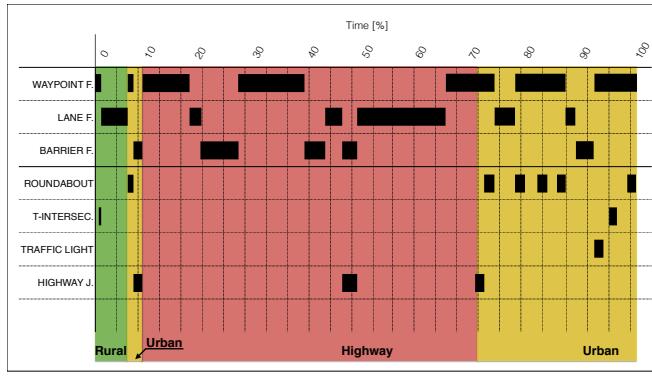


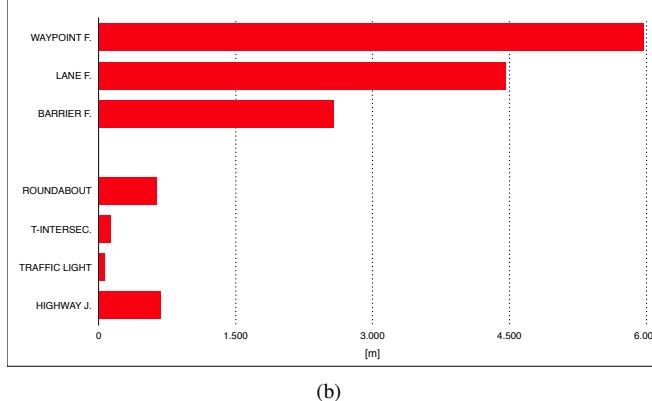
Fig. 8. Speed Profile during the whole PROUD test.

TABLE I
PATH STATISTICS.

Total Distance	13 km
Time of day	11:00 am
Day of week	friday, working day
Urban, suburban environments	2.4 km
Highway environment	10.6 km
Number of roundabouts	6
Number of junctions	3
Number of crosswalks	15
Number of tunnels	8



(a)



(b)

Fig. 9. Maneuvers overview, the environment is specified by the background color: green for rural, yellow for urban and red for highway. (a) Maneuvers selection. (b) Maneuvers usage. The depicted maneuvers are: Waypoint Following, Lane Following, Barrier Following, Roundabout, T-intersection, Traffic-lights, Highway Junction.

km of highways, also outlined in table I. Further details on vehicle behavior are described in figures 8, 9(a), and 9(b).

Figure 9(a) shows the vehicle maneuvers that were engaged by the autonomous vehicle during the test; on the y-axis a list of the possible implemented maneuvers i.e.: GNSS-based Waypoint Following, Lane Following, Roundabout maneuver, T-Intersection maneuver, Traffic Light maneuver, and Highway Junction (those maneuvers are detailed in [15]).

Table II shows some test statistics, highlighting also the GNSS performance.

TABLE II
TEST STATISTICS.

Distance	13 km
Time	18 min
Average speed	43.33 km/h
Max speed	70 km/h
GNSS outage ¹	4 min 22" (20.7%)

To validate and test some algorithms behaviors a set of simulations was used, reporting results like false positives and negatives and comparing algorithms output to a computer generated ground truth. The simulator engine is a separated software provided by TASS International, called PreScan® that is used coupled with Matlab Simulink®.

VI. CONCLUSIONS

This article presented the architecture and main results of the automatic driving test performed in an open urban environment in Parma, Italy. Future articles will dig deeper into each perception and control algorithms.

The vehicle showed to be able to perform autonomous driving together with other vehicles in the same environment, and manage roundabouts, intersections, road priority, stops, tunnels, crosswalk, traffic lights, highways, and urban roads. Maps are only used to trigger the perception subsystems: lane detection, stereo vision, classification and tracking of obstacles demonstrated to be essential.

However, it is important to note that this test – although demonstrating that the vehicle was able to cope with all the challenges of the urban environment in a city like Parma in Italy – is not intended to demonstrate that autonomous urban driving is solved. This experiment constitutes just another step forward in the direction of full automation in a complex environment like the urban one, but many challenges still remain unsolved.

The test was indeed very successful since the vehicle was able to handle all the route (rural, highway, and urban) with no human intervention, but the specific path was previously tested many times and the main challenges were solved as they were met. If a different route was chosen, different challenges would have been highlighted and possibly solved, underlying a lot of space for additional improvement.

Some examples of unsolved challenges are the handling of large roundabouts composed of many lanes or merging into high speed traffic when a short merging lane is present.

Indeed the weather played a basic role in the test; during the test it was sunny therefore no heavy rain or snow had to be considered, although sun shine (like the one in Italy on a July morning) poses some specific issues like reflections and sensor saturation.

The complete test was filmed by many cameras in the vehicle, on a following vehicle, along the road and on a following helicopter. The whole uncut video is available on

¹GNSS outage occurs when the GNSS device runs either in standalone mode nor not enough satellites are available



Fig. 10. A frame of the final part of the test video. Top left: the view of BRAiVE by the following vehicle; top right: the helicopter view. On the bottom, the view of the camera inside BRAiVE and internal data: vehicle speed, steering wheel angle, and selected maneuver.

YouTube and showcases the complete test together with some internal data that show the vehicle's reaction and behavior (see fig.10).

Another very important consideration is that some maneuvers were managed with safety in mind, and sometimes even overly conservative. The vehicle behavior deviated from the common human behavior, especially in roundabouts and mergings. This is an issue that needs to be carefully considered when designing and implementing the next version of those maneuvers, to avoid that the autonomous vehicle behavior be considered too inefficient when compared to a human driver.

Again, further investigations need to cover the handling of unlawful traffic and uncommon behaviors of other road users.

The importance of object classification lies in the fact that more accurate predictions on the objects movement can be made, thanks to this additional knowledge. Definitely understanding and predicting the actions of human road users will be one of the great challenges of the future.

A final consideration is the fact that BRAiVE was developed to be an efficient test bed for autonomous driving and not as a product. Therefore, although safety was one of the main concerns during its development and test, no effort was dedicated in making the system robust to failures. In case of a sensor or PC failure, the system detects it, signals it, but does not operate any emergency maneuver. In other words, the current design does not take into account any redundancy or system robustness.

All the issues mentioned above are challenges for future research.

REFERENCES

- [1] A. Broggi, P. Grisleri, C. Yakes, and C. Hubert, "The Oshkosh-VisLab joint Efforts on UGVs: Architecture, Integration, and Results," in *Procs. of the 146th NATO Wks. on Applied Vehicle Technology panel*, Florence, Italy, May 2007.
- [2] M. Montemerlo, J. Becker, S. Bhat, H. Dahlkamp, D. Dolgov, S. Etinger, D. Haehnel, T. Hilden, G. Hoffmann, B. Huhnke, D. Johnston, S. Klumpp, D. Langer, A. Levandowski, J. Levinson, J. Marcil, D. Orenstein, J. Paefgen, I. Penny, A. Petrovskaya, M. Pflueger, G. Stanek, D. Stavens, A. Vogt, and S. Thrun, "Junior: The stanford entry in the urban challenge," *J. Field Robot.*, vol. 25, no. 9, pp. 569–597, Sept. 2008.
- [3] L. Bombini, S. Cattani, P. Cerri, R. I. Fedriga, M. Felisa, and P. P. Porta, "Test bed for Unified Perception & Decision Architecture," in *Procs. 13th Int. Forum on Advanced Microsystems for Automotive Applications*, Berlin, Germany, May 2009.
- [4] K. Kritayakirana and J. Gerdes, "Autonomous vehicle control at the limits of handling," *Intl. Journal of Vehicle Autonomous Systems, special issue for 10th Anniversary*, vol. 10, no. 4, pp. 271–296, June 2012.
- [5] S. Thrun, "What we're driving at," The Official Google Blog, Oct. 2010, available at <http://googleblog.blogspot.com/2010/10/what-were-driving-at.html>.
- [6] M. Bertozi, L. Bombini, A. Broggi, M. Buzzoni, E. Cardarelli, S. Cattani, P. Cerri, A. Coati, S. Debattisti, A. Falzoni, R. I. Fedriga, M. Felisa, L. Gatti, A. Giacomazzo, P. Grisleri, M. C. Laghi, L. Mazzei, P. Medici, M. Panciroli, P. P. Porta, P. Zani, and P. Versari, "VIAC: an Out of Ordinary Experiment," in *Procs. IEEE Intelligent Vehicles Symposium 2011*, Baden Baden, Germany, June 2011, pp. 175–180, iSSN: 1931-0587.
- [7] Y.-L. Chen, V. Sundareswaran, C. Anderson, A. Broggi, P. Grisleri, P. P. Porta, P. Zani, and J. Beck, "TerraMax: Team Oshkosh Urban Robot," *Journal of Field Robotics*, vol. 25, no. 10, pp. 841–860, Oct. 2008.
- [8] J. Choi, J. Lee, D. Kim, G. Soprani, P. Cerri, A. Broggi, and K. Yi, "Environment-Detection-and-Mapping Algorithm for Autonomous Driving in Rural or Off-Road Environment," *IEEE Transaction on Intelligent Transportation System*, vol. 13, no. 2, pp. 974–982, June 2012.
- [9] Y. Jiang, J. Gong, G. Xiong, Y. Zhai, X. Zhao, S. Zhou, Y. Jiang, Y. Hu, and H. Chen, "Design of a universal self-driving system for urban scenarios 2014; bit-iii in the 2011 intelligent vehicle future challenge," in *Intelligent Vehicles Symposium (IV), 2012 IEEE*, 2012, pp. 506–510.
- [10] A. Broggi, S. Debattisti, P. Grisleri, M. C. Laghi, P. Medici, and P. Versari, "Extensive Tests of Autonomous Driving Technologies," *IEEE Trans. on Intelligent Transportation Systems*, vol. 14, no. 3, pp. 1403–1415, Sept. 2013, iSSN: 1524-9050.
- [11] M. Bertozi, L. Bombini, A. Broggi, P. Cerri, P. Grisleri, and P. Zani, "GOLD: A framework for developing intelligent-vehicle vision applications," *IEEE Intelligent Systems*, vol. 23, no. 1, pp. 69–71, Jan.–Feb. 2008.
- [12] D. Dolgov and S. Thrun, "Autonomous driving in semi-structured environments: Mapping and planning," in *Proceedings of the 2009 IEEE International Conference on Robotics and Automation (ICRA-09)*, Kobe, Japan, 2009, to Appear.
- [13] O. Foundation, "Openstreetmap," available at <http://www.openstreetmap.org>.
- [14] D. Ferguson, M. Likhachev, and A. T. Stentz, "A guide to heuristic-based path planning," in *Proceedings of the International Workshop on Planning under Uncertainty for Autonomous Systems, International Conference on Automated Planning and Scheduling (ICAPS)*, June 2005.
- [15] A. Broggi, S. Debattisti, M. Panciroli, and P. P. Porta, "Moving from analog to digital driving," in *Procs. IEEE Intelligent Vehicles Symposium 2013*, Gold Coast, Australia, June 2013, pp. 1113–1118.
- [16] A. Broggi, A. Coati, P. Medici, M. Panciroli, and P. Zani, "Autonomous vehicles control in the VisLab Intercontinental Autonomous Challenge," *Annual Reviews in Control*, vol. 36, no. 1, pp. 161–171, 2012. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1367578812000132>