Detection, Prediction, and Avoidance of Dynamic Obstacles in Urban Environments

Dave Ferguson, Michael Darms, Chris Urmson, and Sascha Kolski

***Abstract*— We present an approach for robust detection, prediction, and avoidance of dynamic obstacles in urban environments. After detecting a dynamic obstacle, our approach exploits structure in the environment where possible to generate a set of likely hypotheses for the future behavior of the obstacle and efficiently incorporates these hypotheses into the planning process to produce safe actions. The techniques presented are very general and can be used with a wide range of sensors and planning algorithms. We present results from an implementation on an autonomous passenger vehicle that has traveled thousands of miles in populated urban environments and won first place in the DARPA Urban Challenge.**

Fig. 1. Our autonomous vehicle "Boss" during vehicle interaction testing in Pittsburgh, Pennsylvania.

I. INTRODUCTION

Driving in urban environments requires interacting with other vehicles. Whether following behind a slow-moving vehicle, coordinating to take turns with vehicles at intersections, or maneuvering around other vehicles to reach parking spots, it is near impossible to take any voyage in a car without being affected by another vehicle in some manner. As driver assistance systems and autonomous vehicles become more sophisticated, reasoning about such vehicle interactions will become increasingly important. To do so, three capabilities are required. First, other vehicles must be reliably detected, through on-board or off-board sensors or vehicle-to-vehicle communication. Secondly, the future behavior or movement of these vehicles must be imparted or inferred. And finally, this information must be used to provide safe, intelligent courses of action for the driver assistance system or autonomous vehicle. Although much research has been performed in these areas, particularly for robotic systems, current approaches fail to satisfy the requirements of general urban driving.

For dynamic obstacle detection, the configuration of sensors is dependent on the application [1]. Traditional configurations for commercial driver assistance systems couple a single sensor to a tracking model which is in turn tied to a particular application (for instance Adaptive Cruise Control [2]). However, driving in an urban environment with arbitrary road shapes and open areas requires a more general framework for modeling and tracking vehicles.

Further, no single sensor exists that fulfills the requirements for reliable dynamic obstacle detection in urban environments. Strong empirical evidence of this comes from the Urban Challenge Final Event where all of the autonomous vehicles relied on multiple sensors in their perception systems [3]. Unfortunately, using multiple heterogenous sensors increases the complexity of the sensor fusion task, since each sensor has different characteristics that need to be considered to combine their results effectively (see, for example, [4]).

For dynamic obstacle prediction, the simplest approach is to assume that the obstacles remain in their current positions forever and treat them as static. Existing approaches that do treat them as moving often require perfect information about their trajectories [5], [6] or assume they will continue along their current heading and velocity [7], [8], [9]. However, in practice none of these scenarios are realistic: it is unlikely we will have accurate information from another vehicle as to its future trajectory, nor in general will it just continue along its current heading (or stop and sit still). Recently, researchers have extended these approaches to incorporate a notion of uncertainty in the future behavior of other vehicles through probabilistic trajectory models, but these too are heavily biased towards the vehicles continuing their exact current behavior [10]. Sometimes, such models are the best we can do as we have no additional information to draw upon, but in structured environments such as roads and intersections we can exploit this structure to generate much more realistic predictions for the future movement of other vehicles.

Research into generating safe actions amongst moving vehicles has traditionally focused on very simple environments and simple vehicle models [11], [12], [13], or very short-term (often instantaneous) actions [8], [9]. However, effective driving in urban environments can require complex actions

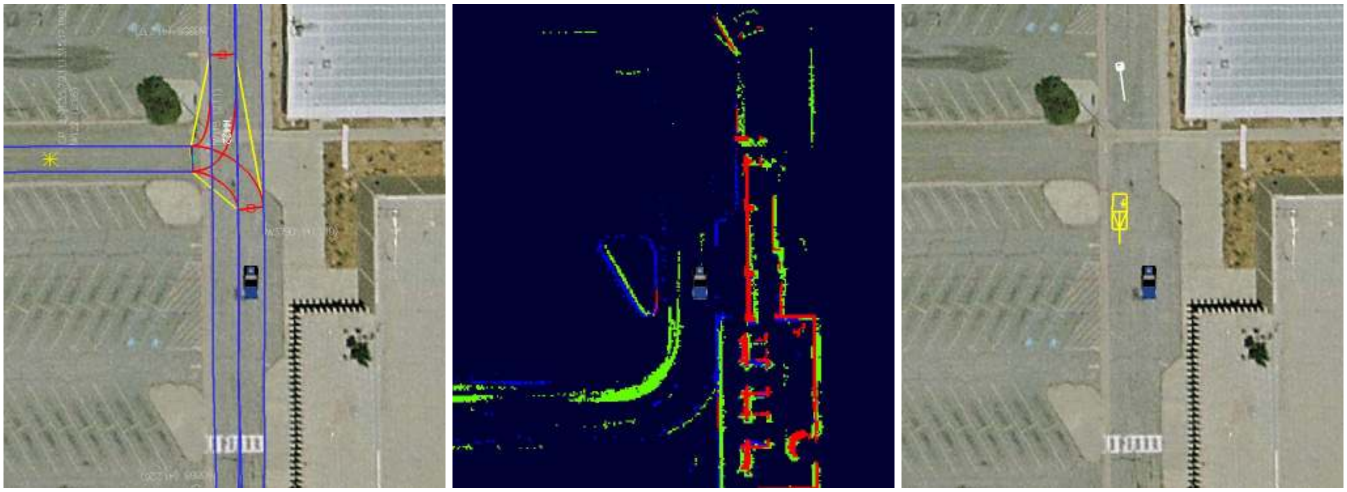
This work would not have been possible without the dedicated efforts of the Tartan Racing team and the generous support of our sponsors including General Motors, Caterpillar, and Continental. This work was further supported by DARPA under contract HR0011-06-C-0142.

D. Ferguson is with Intel Research Pittsburgh and Carnegie Mellon University, Pittsburgh, PA, USA. Email: dave.ferguson@intel.com

M. Darms is with Continental Inc., Auburn Hills, MI, USA. Email: michael.darms@us.contiautomotive.com

C. Urmson is with Carnegie Mellon University, Pittsburgh, PA, USA. Email: curmson@ri.cmu.edu

S. Kolski is with ETHZ, Zürich, Switzerland. Email: sascha.kolski@mavt.ethz.ch



(a) Road Structure (b) Static Obstacle Map (c) Dynamic Obstacle List

Fig. 2. Different Outputs from Perception

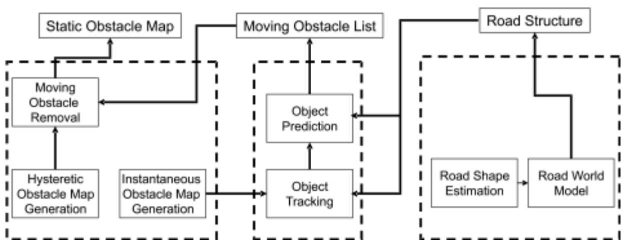


Fig. 3. Perception Architecture.

and non-trivial vehicle models. Further, the number of other vehicles that must be modeled to ensure an action is safe and reasonable can be very large and so efficient methods of reasoning about this interaction are required.

In this paper, we describe an approach for reliable detection, prediction, and avoidance of dynamic obstacles in both on-road and unstructured areas of urban environments. The resulting approach is robust to real-world sensor noise, exploits structure in the environment for realistic prediction of vehicle behavior, and ensures that selected actions are feasible. Further, the approach is general enough to use with a wide range of sensors, vehicle models, and path planners. We also describe an example implementation of the approach on “Boss”, Carnegie Mellon University’s autonomous vehicle entry into the DARPA Urban Challenge, where it has been employed for over 3000 kilometers of autonomous urban driving and contributed to a first place finish in the competition.

II. DYNAMIC OBSTACLE DETECTION

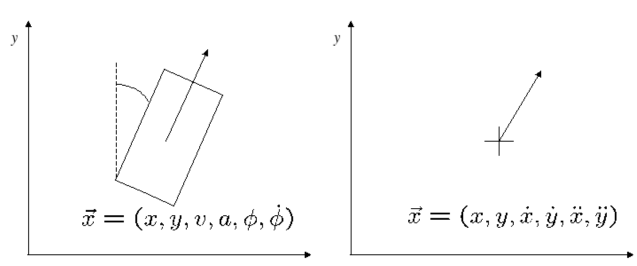
Boss’ perception system provides four principle pieces of information: a vectorized road structure, a static obstacle map, an instantaneous obstacle map and a dynamic obstacle list. Because the dynamic obstacle list is influenced by, and itself influences, the other perceptual outputs we briefly describe all these components.

The road structure is a representation of the lanes and intersections in the environment (in our case, in a vector format – see Fig. 2(a)). This information can be obtained from prior data (such as aerial imagery) and processed in an offline manner or obtained through onboard perception. In our system, we fused both sources to provide an accurate description of the road in the vicinity of the vehicle.

Our static obstacle map representation is a two-dimensional grid (see Fig. 2(b)). Once a dynamic obstacle list is generated, care is taken to remove the dynamic obstacles from this map so that they are not duplicated.

The instantaneous map is very similar to the static obstacle map, but contains all obstacles, static and dynamic. No distinction is made between the two classes and this map is used for target validation in the sensor fusion system for dynamic obstacle detection.

The dynamic obstacle list provides information about all obstacles around the vehicle that are potentially moving. Our dynamic obstacle detection approach represents each dynamic obstacle by an estimation of its shape and its current dynamic properties (see Fig. 5). While our architecture can incorporate an arbitrary number of models, for this application, the shape of each obstacle is one of two models: a box model and a point model (see also [14]).



(a) Box Model (b) Point Model

Fig. 5. Different Dynamic Obstacle Shape Models.

The box model represents the shape of a vehicle while the point model contains no shape information. The point model

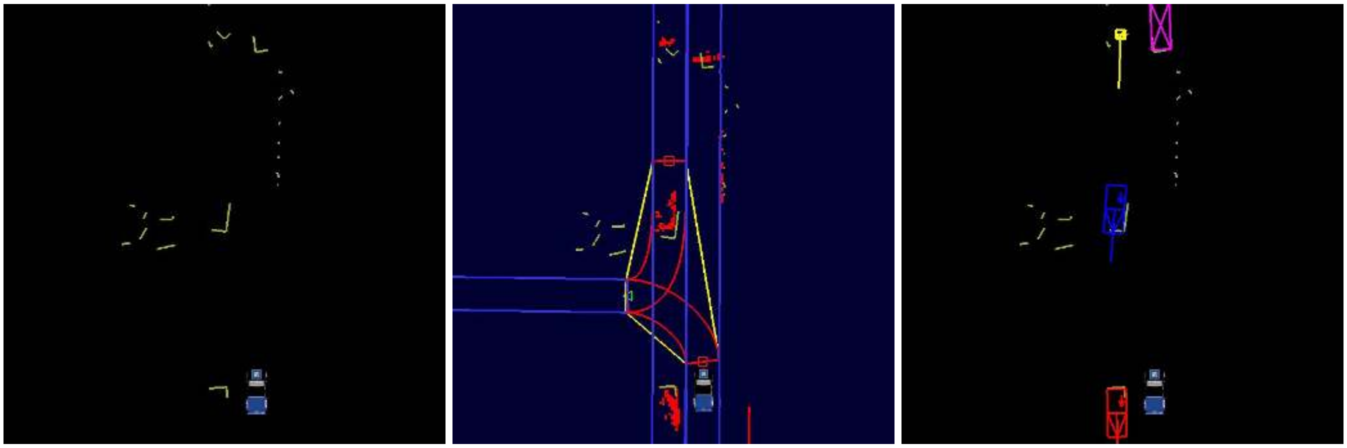


Fig. 4. Detecting dynamic obstacles traveling on roads.

is used when sensor data does not support the box model or when a box representation does not match the features extracted from raw sensor data.

For the box model the velocity and acceleration vectors are always parallel to the longer edge. The orientation is described by an angle ϕ and an angular rate $\dot{\phi}$. The state propagation equations couple the x and y coordinates via the angle ϕ and angular rate $\dot{\phi}$ through a simple bicycle model (see e.g. [15]). The point model is described by two coordinates in the 2D plane and the corresponding velocities and accelerations. For the point model, a constant acceleration model with a gaussian noise component based on the current direction of travel is used for state propagation (see e.g. [16]).

The use of these fixed shape models significantly reduces the complexity of the fusion algorithm (see [14]). In contrast to algorithms using an adaptive or flexible shape model these models do not necessarily represent the actual shape of the tracked object. However, aligning the model with the closest point to our vehicle in general gives a worst case estimation of the position of the tracked vehicle relative to Boss. Empirically, extensive testing showed this approach to be sufficient for on road driving and driving in open parking lots.

To reliably detect dynamic obstacles we use a multi-sensor approach combining radar and laser data from different sensors and sensor technologies. For every sensor a type dependent sensor module is used (see e.g. [17]). Each sensor extracts a set of features and associated them to the current set of dynamic obstacle hypotheses. For example, laser scanner data is processed to extract “L” shaped corner features that could correspond to vehicles.

Each feature is first validated by a sensor specific algorithm. As an example, for features from a radar sensor, the velocity measured by Doppler shift can be used. Next, the features are checked against the instantaneous obstacle map and the road shape to reject false positives (e.g. artifacts caused by ground detections).

Remaining features are then fused into a set of box and point dynamic obstacles. Each sensor module proposes an interpretation of the extracted feature for the best tracking model and a voting algorithm selects the best model for object tracking. Fig. 4 provides an example of the approach in action during an Urban Challenge qualification run. Here, the first image shows the corner features extracted from a planar laser sensor and the second image shows these features being evaluated against the instantaneous obstacle map (map shown in red) and the road. The third image shows the resulting box and point object hypothesis that best explain the sensor data.

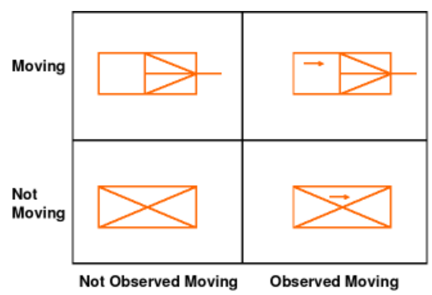
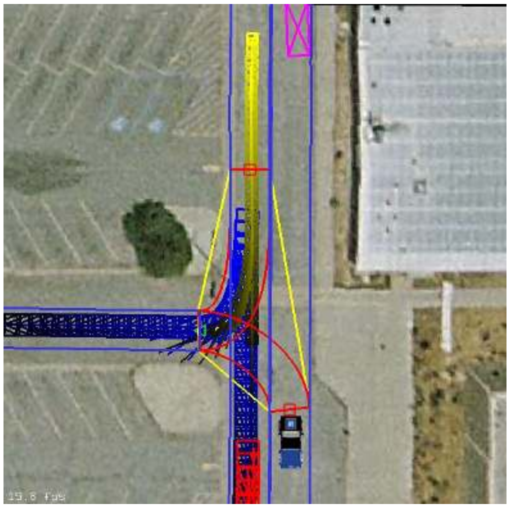


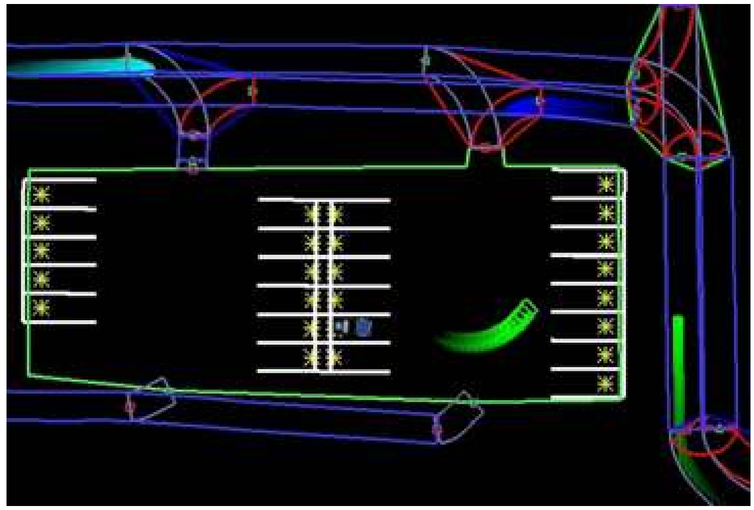
Fig. 6. Different Dynamic Obstacle Velocity Models.

To differentiate between objects that have always been static and may remain static (e.g. parked cars), vehicles that have been moving but are currently stopped (e.g. cars at an intersection), and vehicles that are currently moving, all object hypothesis are further classified into a) *Moving* and *Not Moving* and b) *Observed Moving* and *Not Observed Moving* (see Fig. 6). The *Moving* flag is set if the object currently has a velocity that is significantly different than zero. The *Observed Moving* flag is set when the object has been observed to be moving for a significant amount of time and is cleared when the object has not been detected moving for a prolonged period of time. These durations vary based on the certainty with which the object has been classified as moving.

The consideration of the *Moving* and *Observed Moving* obstacle characteristics removes the need for traditional clas-



(a) Predicting future on-road behavior



(b) Predicting future on-road and parking lot behavior

Fig. 7. Predicting the future behavior of other vehicles on roads and in parking lots.

sification and recognition of vehicles (see e.g. [18]). Further, it allows the planning system the opportunity to treat each of these obstacle classes differently. We also use these flags to decide which of the detected obstacles should be removed from the static obstacle map. Specifically, if an obstacle does not have the *Observed Moving* flag set, we leave it in the static obstacle map and do not treat it as a dynamic obstacle during planning.

III. DYNAMIC OBSTACLE PREDICTION

If an obstacle has been detected as moving, it is important to predict its future motion so that actions can be selected that are safe through time. In general, this prediction problem is extremely difficult, as we do not have control over these other objects so knowing exactly where they intend to go and how they intend to get there is impossible. However, when these dynamic obstacles are vehicles operating in urban environments, it is much easier to infer their likely behavior through exploiting the structure inherent in such environments.

The basic idea is quite simple: vehicles traveling on roads typically follow common rules of the road. For instance, a vehicle driving along a road is most likely to continue driving along the road, and a vehicle at an intersection is likely to choose to travel down one of the roads available at the intersection. This simple idea allows us to generate hypotheses for where a particular vehicle will travel in the future, based on its current behavior and the structure of the environment.

To implement this idea, we first take the detected dynamic obstacle and its position, heading, and velocity. The box model provides an explicit heading estimate, while the point model provides an implicit heading based on the object's velocity. We then take a model of the road structure in the vicinity of the dynamic obstacle and determine which road lane(s) it is currently traveling in. We use the position and heading of the dynamic obstacle to calculate what its current offset is from that lane (i.e. whether it is currently traveling down the center of the lane or is biased to one side). We then hypothesize that the dynamic obstacle will continue to travel down the lane and will likely maintain the same offset that it currently has. However, if the dynamic obstacle is not heading directly down the lane we predict that it will change its heading over time to align itself with the lane. For instance, if a vehicle is entering onto a road it is likely it will align itself with the road.

To provide accurate predictions leading up to intersections and stop-lines, we reason about the future speed of the dynamic obstacle as well as its course. A dynamic obstacle that is approaching a stop-line is predicted to slow down and stop at the stop-line.

If a dynamic obstacle is at or approaching an intersection, we generate multiple hypotheses of where it could go. To do this, we calculate all the possible lanes that it could leave the intersection from and generate hypotheses for each of them using the above approach. Admittedly, this provides a conservative prediction of the future behavior of the vehicle (obviously, it could only actually travel down one of these lanes), but because intersections are typically prone to confusion and accidents, we feel that exhibiting extra caution in these areas is prudent. Fig. 7(a) shows the predicted behavior of the vehicles detected in Fig. 4.

Generating predictions for dynamic obstacles traveling on roads is only part of the solution, however, since urban driving also involves navigating through parking lots and open, unstructured areas. In such scenarios, the structure of lanes and intersections doesn't exist and thus cannot be exploited. Our approach in these areas is to extrapolate the current behavior of the dynamic obstacles, similar to existing approaches mentioned earlier.

However, rather than just using the position and velocity of the dynamic obstacles to perform this extrapolation, the box model allows us to also incorporate the heading and curvature of the obstacle to provide a more accurate

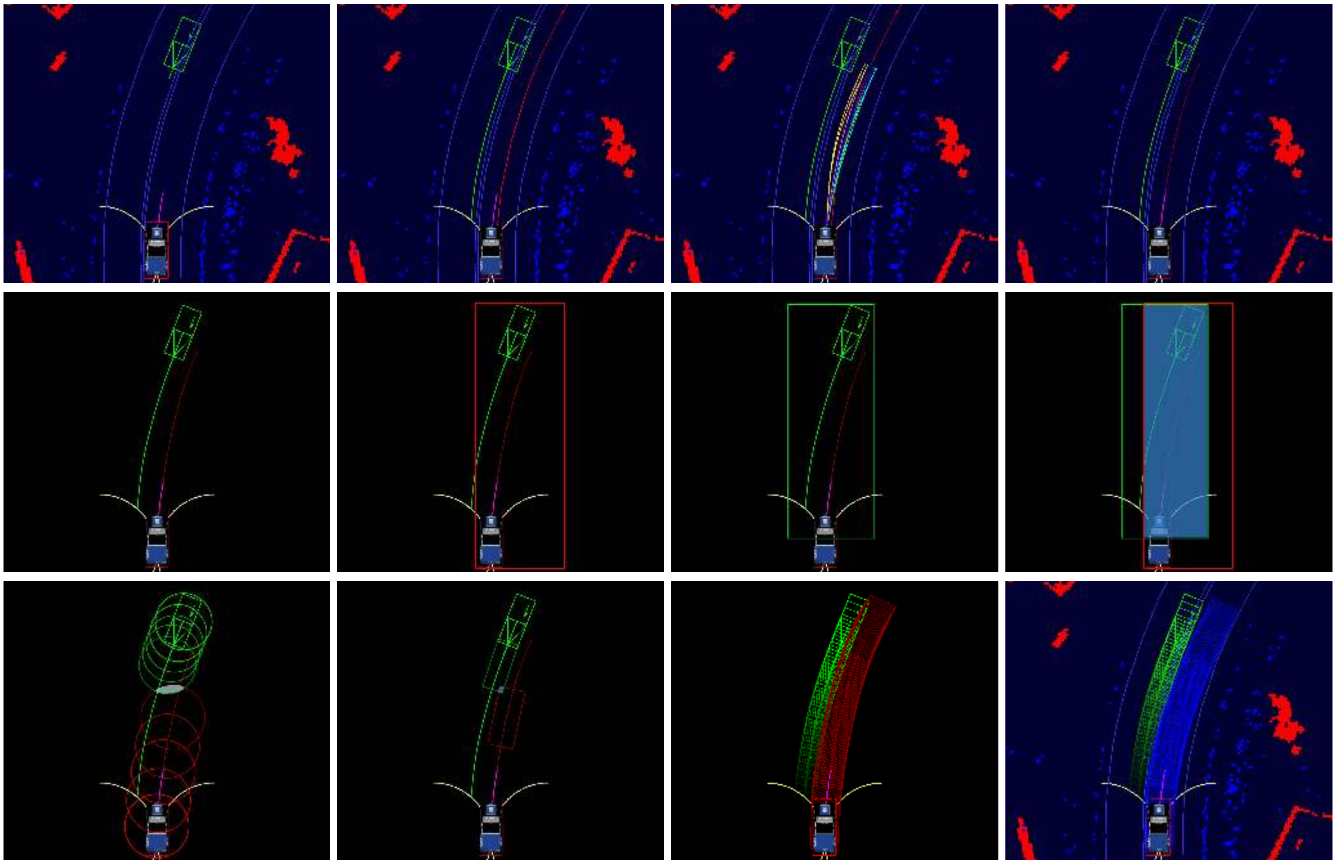


Fig. 8. Following a road lane and avoiding an oncoming vehicle. Our vehicle generates a set of local trajectories down the travel lane and evaluates each to select the best that is collision-free. The steps in the dynamic obstacle collision checking algorithm are shown performed for one of the candidate trajectories (with the surrounding environment removed for clarity). First the worst-case bounding boxes are created for the candidate trajectory and the dynamic obstacle trajectory. Next, since these intersect, the simple pessimistic circles are computed along the trajectories and collision-checked in chronological order. As soon as these intersect at any time frame, the accurate vehicle polygons are collision-checked. In this case, these polygons intersect so the candidate trajectory can be ruled out of contention. This hierarchical approach is equivalent to performing a full check involving the vehicle polygons (shown in the second to last image). The final image shows a different candidate trajectory that does not intersect with the dynamic obstacle and is selected for execution.

short term prediction. For the point model, which is mainly used for dynamic obstacles that are further away from our vehicle, the estimated heading is incorporated but curvature is ignored. Although this prediction model is not as accurate as the on-road model, typically the speeds employed in these unstructured areas are much lower than those on roads, so reacting to updated predictions is much easier and thus the risks of collision are reduced. Fig. 7(b) shows this prediction for a vehicle detected in a parking lot (as well as others detected on the adjacent roads).

IV. DYNAMIC OBSTACLE AVOIDANCE

To safely avoid dynamic obstacles we rely on a motion planner that generates a set of candidate actions for the vehicle and selects from this set one that is collision-free with respect to these obstacles. In our implementation each of these actions is a dynamically-feasible trajectory that can be directly executed by the vehicle. The length of these trajectories varies based on the current speed of our vehicle and is designed to ensure the vehicle could, if necessary, come to a stop over the course of the trajectory. These trajectories are generated using a model-based trajectory generator developed by Howard and Kelly [19] that incorporates a high-fidelity vehicle model to produce an accurate prediction of the vehicle's movement as it executes the trajectory. We can then use this prediction along with our dynamic obstacle predictions to determine whether a candidate trajectory for our vehicle will cause a future collision with any of the dynamic obstacles.

We perform this collision-checking efficiently using a hierarchical approach. Given a candidate trajectory for our vehicle and a predicted trajectory for a dynamic obstacle (extended out in time to match the time duration of the candidate trajectory), we first construct a conservative bounding box for each trajectory. These bounding boxes represent a pessimistic approximation of the area of the environment the trajectories encounter. We then check to see if these bounding boxes overlap: if they don't, then the two trajectories cannot intersect each other; if they do, then there is a chance the trajectories intersect and we must continue to investigate. We then take the two trajectories and step along them in chronological, synchronized time. At each time instant t_i we construct a pessimistic bounding circle of the extent of our vehicle and the dynamic obstacle and check if these circles intersect. If we reach the end of our trajectory without any such intersections, the two trajectories cannot intersect each other[1]. If the circles intersect at some time t_k, then we construct accurate polygonal representations of our vehicle and the dynamic obstacle at this time t_k and check if these polygons intersect. If so, the trajectories will collide with each other and this candidate trajectory is removed from contention. If not, we continue to step forwards in time performing our pessimistic bounding circle checks. We continue in this fashion until we reach the end of the candidate trajectory.

This approach is significantly more efficient than performing the full polygonal collision-checking for every candidate trajectory and dynamic obstacle pair, as the bounding rectangle and circle checks are much less computationally expensive than the polygon intersections. However, the accuracy of the approach is identical to the accuracy of performing full polygonal collision-checking.

Fig. 8 provides an example of the approach in action during the Urban Challenge. In this example, the future path of the dynamic obstacle (in green) is predicted to follow its lane, and the centerline of our vehicle's lane (shown in red, second image in top row) is used to generate a set of candidate trajectories that follow the lane while providing local maneuverability (candidate trajectories are shown in multiple colors in the top-right image). Each of these trajectories is then checked against the static and dynamic obstacles in the environment. The steps in our hierarchical dynamic collision-checking approach are shown in sequence.

As well as being used to rule out candidate trajectories, the existence and predicted behavior of dynamic obstacles can be used to modify the high-level planning of our vehicle. For instance, in unstructured environments such as parking lots, although there are not always lanes to provide guidance, it is common to keep to the right (or left, in commonwealth countries) of other vehicles. If another vehicle is detected and predicted to interfere with some of our candidate trajectories, we can modify the behavior of our vehicle to generate different candidate trajectories that are offset to the right of the other vehicle. This approach was used by our autonomous vehicle in the Urban Challenge to produce safe, considerate driving amongst other vehicles (both robotic and human-driven).

V. CONCLUSION

We have described an approach for reliable detection, prediction, and avoidance of dynamic obstacles in both on-road and unstructured areas of urban environments. Our approach is robust to real-world sensor noise, exploits structure in the environment for realistic prediction of vehicle behavior, and ensures that selected actions are feasible. We have implemented it on an autonomous passenger vehicle and have found it to be very effective over the course of several thousand kilometers of testing. Future research will investigate how this approach can be adapted to commercial driver assistance systems with a human-driven vehicle. In particular, the approach seems well suited to intersection assistance systems, where the road structure and features in the environment can be used to provide prior information for intelligent prediction. Our testing thus far has shown that the presented collision avoidance approach can be effectively used in these and other scenarios as an additional safety layer below higher-level reasoning algorithms.

[1] We assume the time-step used for stepping along the trajectories is sufficiently small (in our case we set it to correspond to a distance of 0.2m along the candidate trajectory)

REFERENCES

[1] R. Bishop, *Intelligent Vehicle Technology and Trends*, Artech House Publishers, London, 2005.

[2] H. Winner, "Adaptive Cruise Control," in Jurgen, R. K. (Hrsg.): Automotive Electronics Handbook. McGraw-Hill, New York, London, 1999.

[3] DARPA Urban Challenge Website: www.darpa.mil/grandchallenge, 2007.

[4] D. Stüker, "Heterogene Sensordatenfusion zur robusten Objektverfolgung im automobilen Straenverkehr." Oldenburg, Univ., Diss. (Online Publication: http://deposit.d-nb.de/cgi-bin/dokserv?idn= 972494464i), 2004.

[5] T. Fraichard and A. Scheuer, "Car-like robots and moving obstacles," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 1994.

[6] Z. Shiller, S. Large, and F. Seckavat, "Motion planning in dynamic environments: Obstacles moving along arbitrary trajectories," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2001.

[7] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using velocity obstacles," *International Journal of Robotics Research*, vol. 17, pp. 711–727, 1998.

[8] T. Fraichard and H. Asama, "Inevitable collision states: a step towards safer robots?" in *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2003.

[9] E. Owen and L. Montano, "A robocentric motion planner for dynamic environments using the velocity space," in *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2006.

[10] C. Fulgenzi, A. Spalanzani, and C. Laugier, "Dynamic obstacle avoidance in uncertain environment combining PVOs and occupancy grid," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2007.

[11] P. Fiorini and Z. Shiller, "Time optimal trajectory planning in dynamic environments," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 1996.

[12] R. Kindel, D. Hsu, J. Latombe, and S. Rock, "Kinodynamic motion planning amidst moving obstacles," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2000.

[13] D. Hsu, R. Kindel, J. Latombe, and S. Rock, "Randomized kinodynamic motion planning with moving obstacles," *International Journal of Robotics Research*, vol. 21, no. 3, pp. 233–255, 2002.

[14] M. Darms, P. Rybski, and C. Urmson, "An Adaptive Model Switching Approach for a Multisensor Tracking System used for Autonomous Driving in an Urban Environment," in Steuerung und Regelung von Fahrzeugen und Motoren - AUTOREG 2008. VDI-Berichte, Dsseldorf: VDI-Verlag, 2008.

[15] N. Kaempchen et al., "IMM object tracking for high dynamic driving maneuvers," in *Proceedings of the IEEE Intelligent Vehicles Symposium*, 2004.

[16] Y. Bar-Shalom and X. Li, *Multitarget multisensor tracking - principles and techniques*, YBS, ISBN 0-9648312-0-1, 1995.

[17] M. Darms and H. Winner, "A modular system architecture for sensor data processing of ADAS applications," in Proceedings of the Intelligent Vehicles Symposium, 2005.

[18] C. Tzomakas and W. von Seelen, "Vehicle Detection in Traffic Scenes Using Shadows," Institut fur Neuroinformatik, Ruhr-Universitat Bochum: Internal Report IRINI 98-06, 1998.

[19] T. Howard and A. Kelly, "Optimal rough terrain trajectory generation for wheeled mobile robots," *International Journal of Robotics Research*, vol. 26, no. 2, pp. 141–166, 2007.