Autonomous Navigation in Crowded Campus Environments

Z. J. Chong*, B. Qin*, T. Bandyopadhyay‡, T. Wongpiromsarn‡, E. S. Rankin¶,
M. H. Ang Jr.*, E. Frazzoli§‡, D. Rus§‡, D. Hsu*, K. H. Low*

* National University of Singapore,
Kent Ridge, Singapore

‡ Singapore-MIT Alliance for Research and Technology,
Singapore

§ Massachusetts Institute of Technology,
Cambridge, MA., USA.

¶ DSO National Laboratories,
Singapore

Abstract—**This paper considers autonomous navigation in crowded city environments. An autonomous vehicle testbed is presented. We address two challenges of pedestrian detection and GPS-based localization in the presence of high-level buildings. First, we augment the localization using local laser maps and show improved results. A pedestrian detection algorithm using a complementary vision and laser system is proposed. We implement this algorithm in our testbed and evaluate its performance. We also show how utilizing existing infrastructural sensors can improve the performance of the system. Potential applications of this work include fully automated vehicle systems in urban environments typical in megacities in Asia.**

I. Introduction

One of the long-standing research activities in robotics has been towards increasing the level of autonomy in both manned and unmanned systems. In transportation systems, automation has been employed, for example, in traffic light management and congestion avoidance services and has attracted numerous research interests in the transportation science and logistics community. Intelligent vehicle/highway systems (IVHS) technologies have also been developed to enhance operational safety and efficiency [13].

In this paper, we focus on putting more autonomy into the transport vehicles to deal with the problem at a more local level. Interests in autonomous driving, especially in urban environments, were largely stimulated by the launch of the 2007 DARPA Urban Challenge (DUC). In this competition, the autonomous vehicles have to navigate, in a fully autonomous manner, through a partially known urban-like environment populated with (static and dynamic) obstacles and perform different tasks such as road and off-road driving, parking and visiting certain areas while obeying traffic rules. As the emphasis of the competition is geared more towards military applications, the vehicles have to be fully self-contained in every aspect including sensing, perception, computing, power, and control. Although this leads to an elegant setup, the situations encountered in DUC do not closely represent those faced in a real-world crowded city like Singapore, London, etc. In addition, the cost of hardware components on these autonomous vehicles is extremely high, making them impractical to be employed in social or commercial applications. The majority of the cost comes from expensive, high-performance sensors (e.g. Velodyne LIDAR) and localization units (e.g. Applanix Intertial Navigation System) needed so that the vehicles can effectively handle all the possible (even adversarial) environments they may encounter.

This paper focuses on crowded city situations where automation can significantly improve the throughput as compared to on ground human decisions. As opposed to adversarial environments faced in military applications, city environments faced in social or commercial applications are typically equipped with infrastructure including cellular networks, traffic cameras, loop detectors and ERP (Electronic Road Pricing) gantries. In addition, the detailed road network and many features of the environment in which the vehicles operate can be obtained a priori. In fact, autonomy in transport vehicles has been attained at various levels of interaction with the infrastructure. Systems that are driven with heavy infrastructure dependence such as mono-rails typically require significant setup and are more suitable for a fully controlled environment such as warehouses and docks. FROG (Free Ranging on Grid) Navigation Systems[1] is an example of such heavily structure-dependent systems and has been deployed for container handling operations in factory sites. Their system consists of a team of autonomous vehicles and a centralized supervisory system. The autonomous vehicles localize themselves using magnets on the floor and GPS for point to point navigation and use laser and sonar for obstacle avoidance. An advantage of this approach is that the global viewpoint of the environment can be obtained while the main disadvantages include delayed information, communication burden and cost of specialized infrastructure setup.

On the other hand, Google driverless car, for example, is fully autonomous [8]. Their converted Toyota Prius uses cameras to detect traffic lights and radars to detect pedestrians. Their autonomous navigation also relies on laser range finders, cameras, radars, inertial sensors and high-detail maps. An advantage of these fully autonomous systems is the ability to handle dynamic and adversarial scenarios. However, as opposed to heavily structure-dependent systems, only limited, local viewpoint can be obtained as on-board sensors on each

[1] http://www.frog.nl/

vehicle have the same limitation of limited viewpoint and occlusion by urban structures and other vehicles. Each vehicle also bears the burden of sensing and perception. Additionally, being fully autonomous and self-contained comes with a heavy price tag.

In this paper, we investigate the integration of the heavily structure-dependent and the fully autonomous approaches in order to build an autonomous vehicle system at reasonable cost as well as keep the high level of autonomy even in a crowded scenario. As an initial step, we focus on campus environments. We exploit an abundance of infrastructural sensors available to the road network that can provide very important information about the presence of other entities on the road in real time. This information can help in planning collision-free optimal trajectories for each vehicle beyond visual range. Exploiting the infrastructure also helps reduce the sensing burden on the vehicle and reduce the cost of the vehicle, making such systems more economical and accessible to normal people. The main contributions of the paper are the following.

1) Typical localization techniques heavily depend on GPS (Global Positioning System). These techniques do not work when dealing with high-level buildings in city environments. We augment the localization using local laser maps and show improved results.
2) We implement a more robust pedestrian detection system using a complementary vision and laser system. We show that such a set up works well even for a moving vehicle.
3) Our approach is to incorporate existing infrastructural sensors whenever possible into the motion planning of the autonomous vehicle. To demonstrate the setup, we show a scenario incorporating the infrastructural sensors that significantly improves the performance of the vehicle operation.

The remainder of the paper is organized as follows. An overview of our testbed is provided in Section II. Section III describes our dynamic obstacle detection component and, in particular, the pedestrian detection algorithm implemented in our testbed. Its performance is evaluated, both for single and multiple pedestrian cases. The localization and navigation components are described in Section IV. Section V describes the role of infrastructural sensors in our system. Finally, Section VI concludes the paper and discusses future work.

II. Autonomous Vehicle Testbed

Our autonomous vehicle testbed is a Yamaha G22E golf car mounted with various sensors (Fig.1a). It was modified and equipped with actuators to achieve drive by wire capabilities. Two servo motors are used to control the steering angle and amount of brake applied separately. Since it is an electric vehicle, the throttle control is readily accessible through a varying PWM voltage signal that can be regulated by a low level controller. To fulfill the power requirement for a wide variety of sensors, a 1350W inverter was used. For the sensors, a wheel encoder is fitted at the left front wheel. The steering angle and brake are inferred implicitly from the motors encoder. To receive GPS signal, Ublox EVK-6R is used. The module comes with Enhanced Kalman Filter to give an estimate of global location using integrated data input from wheel encoders tick count and onboard gyroscope. The main sensors are the laser range finders that consist of two SICK LMS 291 and Hokuyo UTM 30LX. The SICK lasers have a range of 80 m with 180° field-of-view (FoV). The Hokuyo sensor, on the other hand, has 270° FoV with 30 m range. A normal webcam is fitted on one of the SICK lasers to provide visual feedback and to perform vision processing.

Software Architecture: We have developed a modular software architecture for ease in incorporating additional functionality without modifying the core system. The system has been integrated into ROS (Robotic Operating System), which provides a standard form of communication among different software modules [10]. The main modules that have been implemented on the current system include perception, mapping-and-prediction, localization, planner and controller as shown in Fig. 1b.

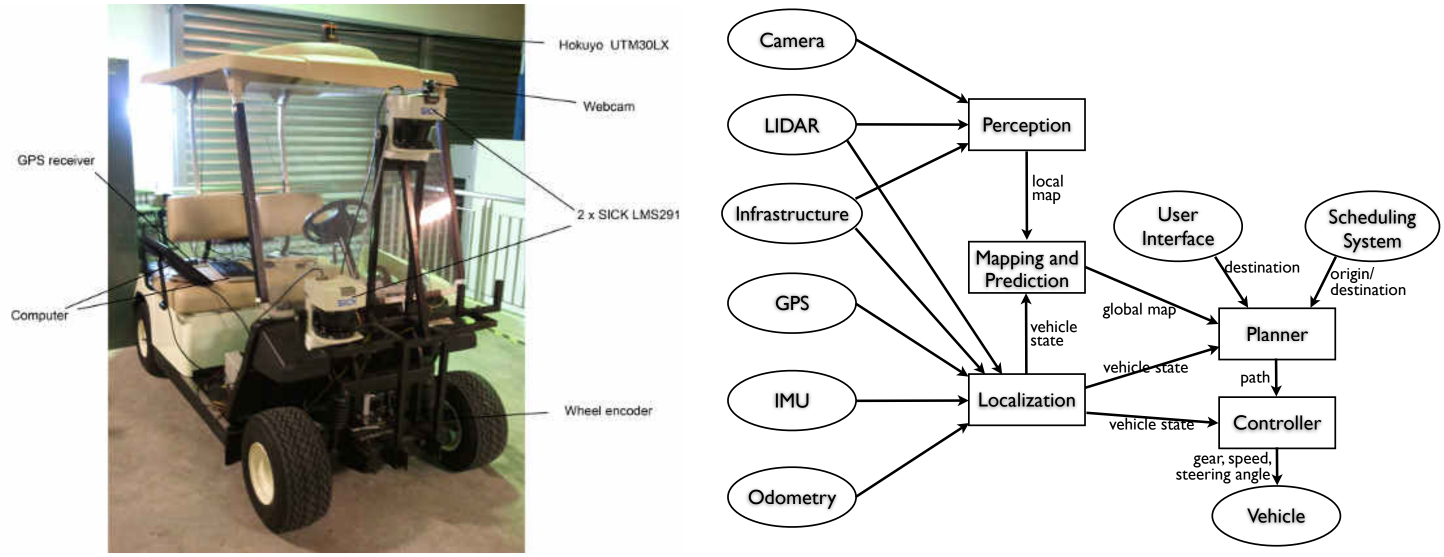
The perception module takes as an input the raw sensed data from the sensors. Detection and tracking algorithm is then applied to extracted features (e.g., pedestrian and other moving and stationary objects) from the raw sensed data. As discussed in Section V, the sensors we have utilized include not only onboard cameras and laser range finders but also infrastructure cameras installed, for example, at an intersection. The data from these infrastructure cameras are transmitted through a WiFi network. To reduce the amount of data that needs to be transmitted, the raw sensed data may be processed so that only important features (e.g., other vehicles and pedestrians approaching the intersection) are transmitted.

The mapping-and-prediction module maintains the global map of the environment in which the system is operating. In addition, it predicts the motion of moving objects such as pedestrians. The localization module incorporates the data received from the GPS (Global Positioning System), IMU (Inertial Measurement Unit), laser range finder and vehicle's odometer to provide an estimate of the current state (position, velocity and heading) of the vehicle.

The planner module is responsible for computing an obstacle-free path, satisfying certain traffic rules, to the goal. In the case where the user is onboard, the goal may be specified as the destination through the user interface. Alternatively, the scheduling system that computes the pick-up and drop-off position for each autonomous vehicle may send the origin and destination to the system through a cellular network. Finally, the controller module is responsible for computing the actuator commands, including the speed, gear and steering angle, to the physical actuators so that the vehicle closely follows the planner-generated path.

III. Dynamic Obstacles Detection

For autonomous navigation, we need to pay special attention to dynamic objects, like pedestrians and other vehicles on the road in addition to static environmental features like kerbs, drains, traffic lights, *etc* (Fig.2). Usually the presence of



(a) System Testbed (b) System Architechture

Fig. 1. Autonomous Vehicle Testbed

(a) (b)

Fig. 2. Pedestrians and other dynamic vehicles need to be detected and handled. (a) NUS campus road, (b) Golf-cart operating in the presence of pedestrians

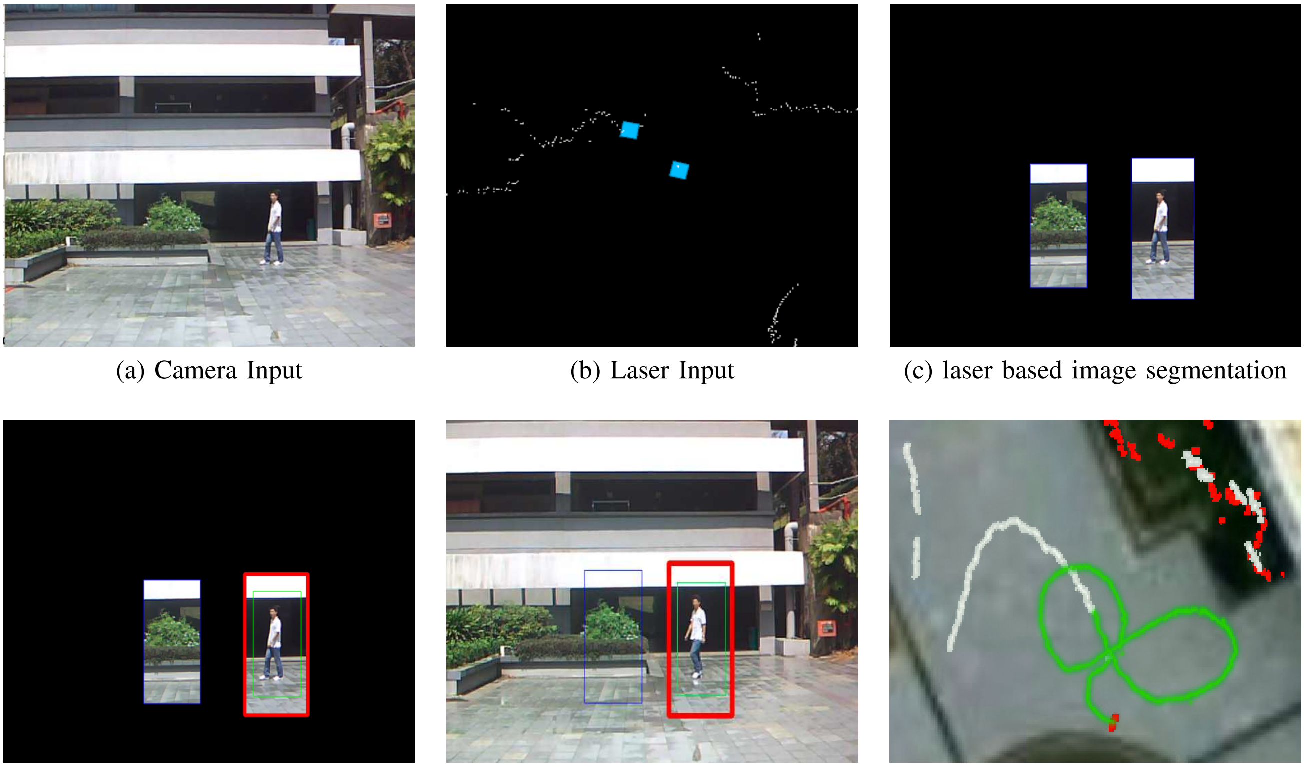
static objects are known a-priori from a traffic database or built during and initial phase in an offline manner. However, dynamic objects on the road can only be handled while the autonomous vehicle is driving. Pedestrians, as a key factor in a campus environment, deserve more attention. On-board cameras are one of the most effective ways of identifying objects in the environment. However, the computation requirement and dependence of ambient light conditions limit their utility. Alternatively, laser based approaches can detect the presence of an object more reliably but have problem disambiguating different types of objects. In our project, we built an onboard pedestrian detection system by hierarchical fusing of a single-layer LIDAR and a simple off the shelf webcam. We combine the advantages of LIDARs in detecting an object with the simplicity of disambiguating objects from the camera images. It proves to be fast, computationally efficient and robust in our operations.

Significant research has been done on pedestrian detection and tracking with LIDARs and vision. The LIDARs provide accurate position information of scanned points in the surroundings, which will be segmented, clustered, and then classified into different objects. Pedestrians can be extracted with certain criteria or features, such as static features of shape and size founded in [5], [9], or dynamic features of gait founded in [14], [2], and so on. These algorithms perform well with multiple LIDARs placed off-board and in relatively structured environment, but would probably fail in real urban traffic, due to severe occlusion and complex surroundings. In the final analysis, limitation of these algorithms comes from sparsity of information of LIDARs. The idea of multi-sensor fusion arises to counter this limitation. The most common type that can be found is a combination of LIDAR and camera. While some related algorithms have been introduced in [4], [6], few of them are suitable to autonomous vehicles, with considerations to the demanding working environment. An algorithm similar to our approach is proposed in [3]. It depends on a four-layer LIDAR to track pedestrians and do preliminary classification, and then use camera to refine the classification belief. In our project, a similar algorithm is proposed. While we also rely on a single-layer LIDAR to track objects, we do not try to classify them in this part, but leave that to the following part of vision verification. Our algorithm proves fast, computationally efficient, robust in operation, and easy to implement.

A. *Pedestrian detection algorithm*

In our implementation the moving object tracking is realized with a single-layer LIDAR. While the approach is general to any dynamic object, like vehicles, pedestrian and other objects, we take the pedestrians as a representative class to talk about in details in this paper. Fig.3 show the flow of the algorithm while, Fig.4 shows the result of the detection algorithm for a single data frame. The algorithm runs in two phases, *pedestrian candidate detector* and *pedestrian verification*.

Pedestrian candidate detector: The LIDAR data is segmented and clustered based on their position and relative velocity (Fig.4b). Potential candidate clusters for pedestrians are filtered out based on their size and velocity. We use a



(a) Camera Input

(b) Laser Input

(c) laser based image segmentation

(d) HOG classifier output

(e) Robust pedestrian detection

(f) Track of single pedestrian.

Fig. 4. Pedestrian detection module

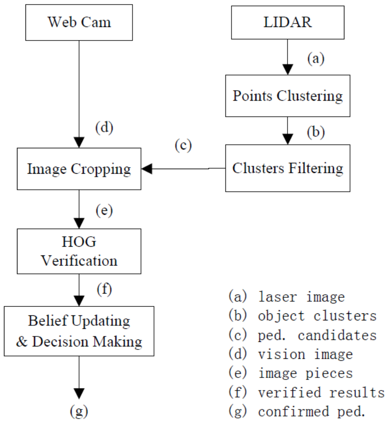


Fig. 3. Pedestrian detection module.

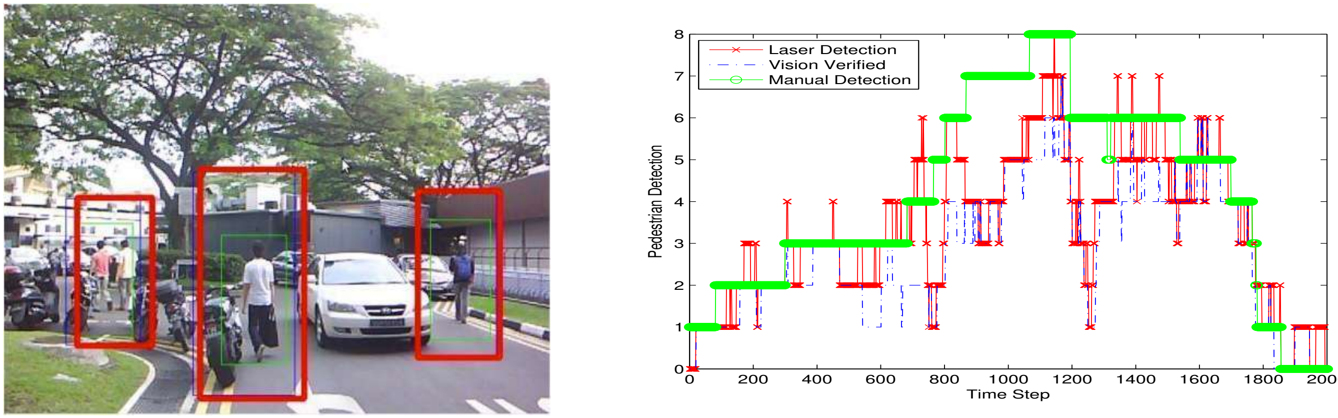
simple linear velocity model in our implementation. However advanced model checks could also be used for higher accuracy. In fact keeping a more relaxed and conservative filter decreases the rate of false negatives in the subsequent pedestrian verification phase.

Pedestrian verification: In this part, we use a common webcam to verify whether extracted objects are pedestrians or not. Extrinsic calibration of webcam and LIDAR is done beforehand. These candidates are projected onto certain areas of webcam image correspondingly. The whole image is then cropped into several smaller sub images (Fig.4c). Since only a small number of sub-images are processed, we decrease the computational time in image processing significantly. The vision verification algorithm used here is histogram of oriented gradient object detection (HOG). In this work, a default trained people detector Support Vector Machines (SVM) from OpenCV was chosen. To enable fast verification, GPU accelerated HOG algorithm was used. HOG classifier identifies each sub image containing pedestrians (Fig.4d) and we label the LIDAR tracks accordingly. This helps us in avoiding running vision based pedestrian detectors on the whole image (Fig.4a,e) and significantly reduces the computational load. The reduced computation allows us to run such detectors in real time on the vehicle.

Note that the pedestrian detection allows us to improve the motion planning for the autonomous vehicles by reasoning about the motion models of the pedestrian obstacles. In the case of false negatives due to vision errors and FoV limitations, the pedestrian clusters are still tracked by the laser, treated as a generic dynamic object and avoided accordingly.

B. Performance evaluation

Single pedestrian detection: Fig.4 shows how this system detects and track a single pedestrian. Fig.4(f) shows the track of this pedestrian. At first, the pedestrian gets tracked by LIDAR, and labeled as a dynamic object shown in white. After it enters the FoV of webcam, it gets verified as a pedestrian and the track turns green. Other potential dynamic objects in the image are correctly rejected.



(a) Snapshot of the onboard pedestrian detector

(b) Pedestrian detection reliability during the autonomous run

Fig. 5. Pedestrian detection from an autonomous run of our testbed.

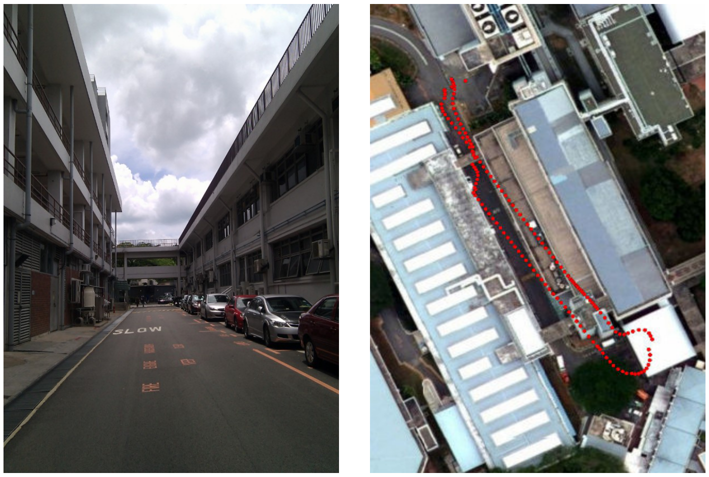
Multiple pedestrian detection: Fig.5 shows the result of our pedestrian detection module while the autonomous vehicle was executing a run. Fig.5a shows the verification of multiple pedestrians. We see that when the pedestrians are too close to each other the laser signatures get merged and they are detected as a single cluster. However, in the view of motion planning it does not matter how many actual pedestrians are close by, our autonomous vehicle avoids the pedestrians effectively. Fig.5b shows the number of objects tracked by LIDAR, pedestrians verified by webcam, and the ground truth number of pedestrians. Because FoV of the webcam is much smaller than LIDAR, objects are accounted only after they can be seen by vision. In the test, our vehicle drived in a really cluttered environment, pedestrians passing by it were tracked and detected. When few pedestrians appeared, they could be easily tracked by LIDAR, and further verified by webcam. However, when there were large numbers of pedestrians causing severe occlusion, pedestrians far away inevitably got lost. At the same time, because a tight group of pedestrians were counted as one, sometimes the number of verified pedestrians appeared to be fewer than the ground truth. In the test, most pedestrians got detected, whether as an individual, or as a group, making safe autonomous driving of our vehicle. Frequency of this detection system is up to 37Hz, limited by scan frequency from LIDAR. Range of effective detection is about 15 meters, limited by resolution of webcam.

Through experiments in different conditions, our onboard pedestrian detection system proves to be both efficient and reliable, and at same time, easily tractable. But due to limited FoV and poor resolution of common webcam, pedestrians walking aside or too far away cannot be verified. At the same time, because we take group of pedestrians as a single one, we cannot get the exact number of pedestrians.

IV. Localization and Autonomous Navigation

Most of the popular approaches in autonomous navigation outdoors depend heavily on GPS based localization. In fact the DARPA challenge was based on GPS based waypoints as input. However GPS is not very reliable in urban areas due to multi-path, limited satellite view in tall sky-scrapers. Such a scenario is shown in Fig.6a. A cloudy sky coupled with tall buildings can attenuate the GPS signals, resulting in erroneous localization as shown by the GPS track in Fig.6b. As the vehicle moves, its GPS erroneously estimates the vehicle location inside buildings and a pure GPS based localization and control could lead to failed navigation.

Interestingly one of the main reasons of GPS limitation i.e, the proximity of buildings, itself provides a good opportunity to utilize range based localization algorithms. In our work we use the laser based maps to augment in regions where GPS underperforms.



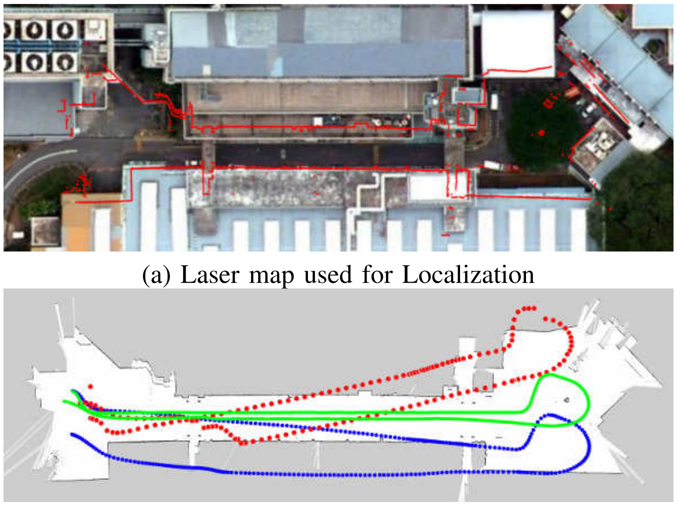
(a)

(b)

Fig. 6. Bad GPS with integrated gyro based odometry

A. Localization

We evaluated several localization approaches on our autonomous vehicle testbed in the campus environment. While the integrated GPS + Gyroscope localizes the vehicle in a global manner, as shown above (Fig.6b), it gives quite poor position estimate. Another approach we looked into was using the LIDAR corrected dead reckoning. Odometry information derived from wheel encoder was able to give an accurate estimation about the distance travel by the vehicle. To obtain the vehicle heading, scan matcher technique was used by comparing the laser scan return. By combining these 2 estimates, we obtained a well-established estimate of the vehicle state. For an autonomous vehicle operating in an urban or campus environment, it is reasonable to assume that a-priori maps could be generated to aid online navigation. With the availability of a-priori laser map, Adaptive Monte Carlo Localization (AMCL) [11] technique was also evaluated. A prior map was built by using the laser thats mounted at a height of 1.83 m. The height of the laser was chosen such that more stable features can be collected and thus increase the belief of the position of the vehicle even in a dynamic environment. To build the map, the vehicle was driven around the environment at a slow speed. While the vehicle was going around, raw laser data and odometry information collected. Generation of map was done offline using SLAM technique that is available from OpenSLAM [1]. Similar to map building, information from odometry and raw laser data was used as input data to provide the most recent observation about the environment and perform calculation on the most probable location of the vehicle.



(a) Laser map used for Localization

(b) Results of various localization algorithms

Fig. 7. Comparison of various localization schemes

Laser based corrections significantly improve on the pure GPS based localization. We show the results of an autonomous run in Fig.7. Fig.7b shows the localization results of various algorithms running on the same run. The red track shows the GPS logs while the blue track shows the location of the vehicle using the odometry alone. The green track is computed by the AMCL algorithm. This shows that in areas of poor GPS locations, building a-priori occupancy maps and subsequently localizing using approaches like AMCL significantly improve the operation. For all our runs, we have used the AMCL algorithm which has shown to work robustly even in the presence of temporary spatial occlusions due to other dynamic vehicles.

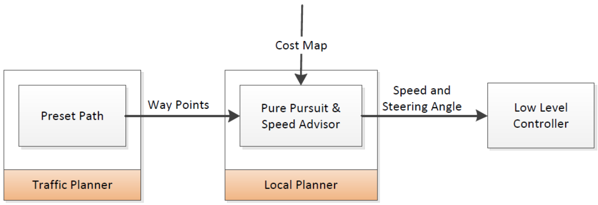


Fig. 8. Navigation module of the autonomous vehicle

B. Cost based Navigation function

Fig.8 shows the structure of our navigation module. During each run, the vehicle maintains its own map based on rolling basis, with the vehicle centered on the map. A map of 50m x 50m with a grid cell size of 0.2m is maintained at all time. Each cell in the map can have a 1 byte value. Initially, the cells in the map are marked as unknown with a value of 255. Whenever an obstacle is observed, the map is updated with a cost value of 254, with the cells now marked as obstacles, the cost is propagated radially outward with an exponential function. At the low level, speed and steering control are separated. For the speed control, the vehicle considers the following input before planning for next action: the average cost function that is present within a defined area in front of itself and the curvature of the path. To ensure stability, a conservative approach is utilized. First, an exponential function is used to calculate the safe speed given the steering angle of the golf car. Then, from the normalized average cost along the projection of the golf car within a fix distance, another safe speed is obtained. Between these 2 values, the minimum one is chosen as the final set point for the speed. The implementation of waypoint follower uses pure pursuit control [7]. Since the golf car's maximum speed is limited to 2 m/s, the look-ahead vector is fixed to 3 meters.

V. Exploiting Infrastructure Sensors

An important feature that distinguishes urban environments from those considered in military applications is the technological advances that we can exploit in order to increase safety and efficiency of the system without imposing much additional cost. Consider, as an example, the scenario where an autonomous vehicle has to traverse an intersection. In many cases, other vehicles approaching the intersection from other directions may not be detected properly by the onboard sensors due to limited sensing coverage and occlusions caused by structures and other environmental features. In [12], the authors mitigate this problem by using two pointable long-range sensors and propose a methodology for computing the optimal angles of the sensors to maximally cover relevant unobserved regions of interest. A method for detecting occlusions is also presented. A phantom vehicle is then placed in the occluded area, leading to a safe but potentially conservative result.

In this work, we consider utilizing infrastructure cameras installed, for example, at an intersection, rather than completely relying on the onboard sensors. These infrastructure cameras can provide information about whether there are pedestrians or other vehicles approaching the intersection. The information can then be transmitted through a WiFi or cellular network. An advantage of this approach is that more accurate information can be obtained as the infrastructure cameras may be mounted to avoid occlusions. In addition, as the number of autonomous vehicles in the system exceed the number of intersections, the cost can be substantially reduced. In fact, in many modern cities, cameras are already installed at many intersections to detect traffic violations. Hence, this approach may incur almost no additional cost.

A. *Avoiding Unobservable Pedestrians*

To show the effect of additional information, we simulate an infrastructural sensor as a wifi-node broadcasting specific information. The infrastructural camera detects the presence of pedestrians and gives a binary information to the golf-cart whether there are pedestrians about to cross the road or if the region is pedestrian free. Currently we are not building models of pedestrian intentions to analyze whether the pedestrian is facing the road or whether s/he is just waiting rather than trying to cross the road. Any pedestrian detection would trigger the autonomous vehicle to slow down in anticipation for the pedestrian to cross the road. The rate of pedestrian detection is 5 Hz. However, since the algorithm only depends on the pedestrian detection alone and not a more detailed analysis based on the pedestrian position and heading, it would also work well with modern traffic/security cameras operating around 1Hz.

Fig.9(a,b) show the view of the scene from onboard as well as a mock infrastructure sensor. The detection of the pedestrian in the left of the image in Fig.9a is quite difficult due to the occlusion from pillars and railings. The autonomous vehicle has to communicate with an existing sensor (security camera) to get more information to plan its path. The pedestrian detection is much easier in Fig.9b. The autonomous vehicle gets the pedestrian information from the infrastructure pedestrian detector sensor and modifies its motion plan, as shown in Fig.9c. We see that during the detection of pedestrians, the autonomous vehicle checks for the possibility of collision and slows down if there is high chance of collision with the pedestrian. The videos of the operation can be accessed at http://dl.dropbox.com/u/20792983/pedestrianVisual1.mp4 and http://dl.dropbox.com/u/20792983/pedestrianVisual2.mp4.

1) Control comparative experiments: Clearly, we would see an improvement in the navigation performance while incorporating the information from an infrastructural sensor. We run a control experiment in simulation to compare quantitative improvement. Our simulator setup is shown as in Fig.11. One or more vehicles move in a loop that has a pedestrian crossing. A pedestrian detection sensor (i.e., a traffic camera) detects the presence of pedestrians and sends information to the autonomous vehicles $V1, V2, \ldots$. Depending on the presence of pedestrians, the vehicles slow down or keep moving. The vehicles are also constrained to maintain a minimum distance between them to avoid collision. We run the experiment both for a single vehicle as well as multiple vehicles. We compare the performance with the baseline case where there is no infrastructural sensors and the autonomous vehicle has to come to a stop before detecting pedestrians on road, something similar to a regular stop and yeild traffic sign. We run the simulation for various vehicle speeds and various number of vehicles. Let T_{base} be the time taken to reach the pedestrian crossing by the baseline algorithm, while $T_{infra.}$ be the same measure for our algorithm getting additional information from the infrastructure sensor. We compute the difference in the time taken to complete each lap, as the time gained by using the infrastructural sensor, $T_{gain} = T_{base} - T_{infra.}$. In both cases the pedestrians appear in a stochastic manner using a poisson process of mean rate of $\lambda = 1 ped/sec.$

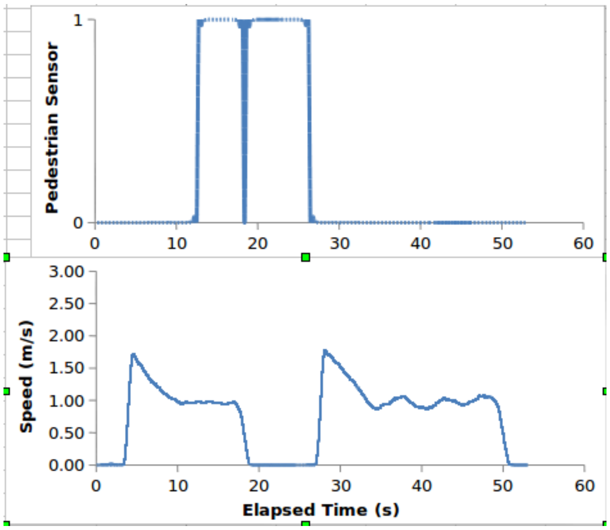


Fig. 10. Velocity profile with pedestrian detection

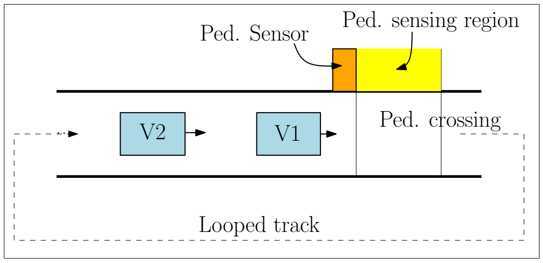
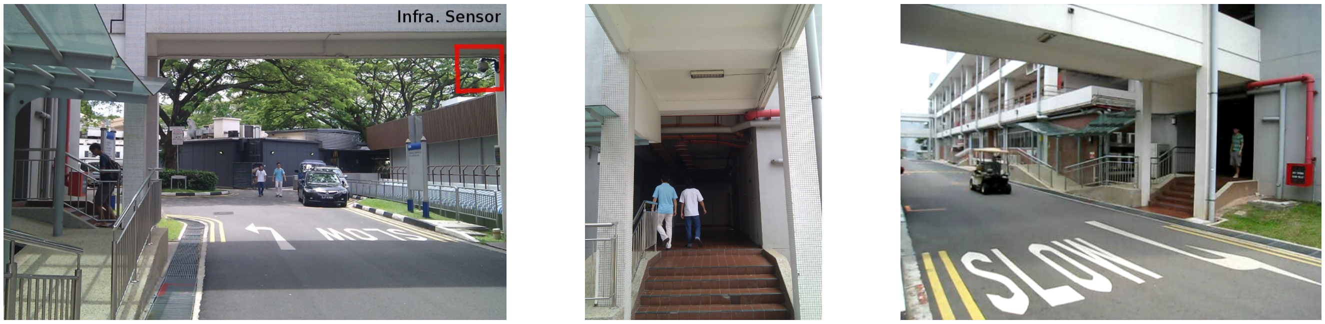


Fig. 11. Simulation environment setup.

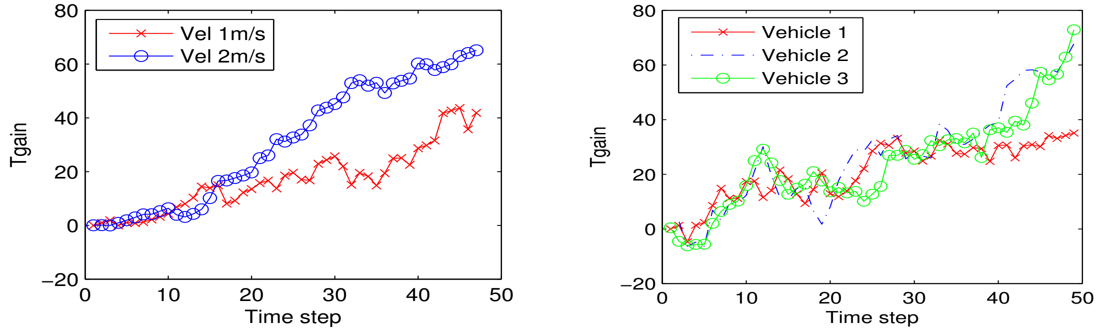
Single Vehicle: Fig.12a, shows the plot of T_{gain} vs the number of laps the vehicle completes. We see clearly that the cumulative time gained by using the infrastructural sensor improves with time. We also note that such a gain is more significant when the vehicle moves at a higher speed. The blue plot is the gain for vehicle moving at 2m/s while the red at 1m/s. This shows that the traffic flow at pedestrian crossings where the vehicles are able to move at higher speeds can be significantly improved by using infrastructural sensors.

Multiple Vehicles: Fig.12b, shows the plot of T_{gain} vs the number of laps the vehicle completes for multiple vehicles. We see that as the number of vehicles increases, T_{gain} also increases. This is because in the baseline algorithm, each vehicle has to stop for pedestrians whether or not they are present. Additionally they have to stop to maintain a minimum distance to the vehicle in front when the front vehicle stops. The number of vehicle stops increase significantly when the number of vehicles increases raising the discrepancy between the proposed and the baseline algorithm.



(a) Onboard camera view (b) Infrastructure camera view (c) Vehicle in operation

Fig. 9. Pedestrian crossing experiment



(a) Single vehicle (b) Multiple vehicles

Fig. 12. Improvement in traffic flow due to incorporating infrastructural sensing

VI. Conclusion and Future Directions

We considered three main challenges in autonomous navigation in crowded city environments: localization, pedestrian detection and limited onboard sensing capability. We showed that in the proximity of tall buildings, popular GPS-based localization can be extremely erroneous. Odometry-based localization was shown to perform slightly better. In order to achieve acceptable performance, we augmented the localization using local laser maps based on Adaptive Monte Carlo Localization technique and showed significantly improved results. We also integrated the use of vision and LIDARs to achieve more robustness in pedestrian detection and tracking. Finally, we exploited existing infrastructural sensors to improve the onboard sensors visibility. The performance of the overall system was evaluated.

Future work targets at augmenting the current system to a fully automated campus vehicle system. To this end, we are currently investigating the use of WiFi-based localization as a complementary approach to GPS-based and laser-based localization. We also plan to incorporate pedestrian intentions in motion planning. Implementation of high-level logics to ensure that the autonomous vehicle obeys traffic rules, properly handles pedestrian and responds to faults and failures is also of interest. Finally, the system needs to be verified for safety both for nominal operations and in the presence of faults and failures.

References

[1] OpenSLAM. http://openslam.org/.

[2] J. Cui, H. Zha, H. Zhao, and R. Shibasaki, "Robust tracking of multiple people in crowds using laser range scanners," in *Proceedings of ICPR*, vol. 4, 2006, p. 857.

[3] F. N. G. Gate, A. Breheret, "Fast pedestrian detection in dense environment with a laser scanner and a camera," in *Proceedings of IEEE Vehicular Technology Conference*, 2009.

[4] R. S. H. Zhao, "A real-time system for monitoring pedestrians," in *Proceedings of IEEE WACV/Motions*, vol. 1, 2005, p. 378.

[5] V. W. K. Fuerstenberg, K. Dietmayer, "Pedestrian recognition in urban traffic using a vehicle based multilayer laserscanner," in *Proceedings of Intelligent Vehicle Symposium*, 2002.

[6] Y. Kobayashi and Y. Kuno, "People tracking using integrated sensors for human robot interaction," in *Proceedings of 2010 IEEE International Conference on Industrial Technology*, 2010, p. 1617.

[7] Y. Kuwata, J. Teo, G. Fiore, S. Karaman, E. Frazzoli, and J. P. How. Real-time motion planning with applications to autonomous urban driving. http://dspace.mit.edu/openaccess-disseminate/1721.1/52527.

[8] J. Markoff, "Google cars drive themselves, in traffic," The New York Times, October 9, 2010.

[9] T. H. O. Mozos, R. Kurazume, "Multi-part people detection using 2d range data," in *Proceedings of 2009 IEEE ICRA*, 2009.

[10] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: an open-source Robot Operating System," in *ICRA Workshop on Open Source Software*, 2009.

[11] D. F. Sebastian Thrun, Wolfram Burgard, *Probabilistic Robotics*. MIT Press, 2005.

[12] Y.-W. Seo and C. Urmson, "A perception mechanism for supporting autonomous intersection handling in urban driving," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2008, pp. 1830–1835.

[13] S. E. Shladover, "Potential contributions of intelligent vehicle/highway systems (IVHS) to reducing transportation's greenhouse gas production," *Transportation Research Part A: Policy and Practice*, vol. 27, no. 3, pp. 207 – 216, 1993.

[14] H. Zhao and R. Shibasaki, "A novel system for tracking pedestrians using multiple single-row laser-range scanners," in *Proceedings of 2005 IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 35, no. 2, March 2005.