Proceedings of the 11th International IEEE
Conference on Intelligent Transportation Systems
Beijing, China, October 12-15, 2008

Vision-based Perception for Autonomous Urban Navigation

Mayank Bansal, Aveek Das, Greg Kreutzer, Jayan Eledath,
Rakesh Kumar, Harpreet Sawhney
Sarnoff Corporation
201 Washington Rd, Princeton, NJ, USA
{mbansal,adas,gkreutzer,jeledath,rkumar,hsawhney}@sarnoff.com

***Abstract*— We describe a low-cost vision-based sensing and positioning system that enables intelligent vehicles of the future to autonomously drive in an urban environment with traffic. The system was built by integrating Sarnoff's algorithms for driver awareness and vehicle safety with commercial off-the-shelf hardware on a robot vehicle. We implemented a modular and parallelized software architecture that allowed us to achieve an overall sensor update rate of 12 Hz with multiple high resolution HD cameras without sacrificing robustness and in-field performance. The system was field tested on the Team Autonomous Solutions vehicle, one of the top twenty teams in the 2007 DARPA Urban Challenge competition. In addition to enabling autonomy, our low-cost perception system has an intermediate advantage of providing driver awareness for convenience functions such as adaptive cruise control, lane departure sensing and forward and side-collision warning.**

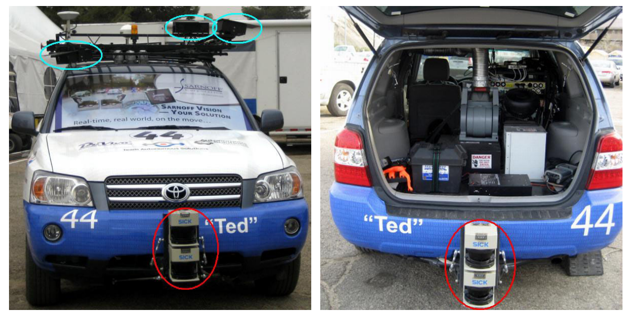


Fig. 1. “Ted” with sensors highlighted. Blue ovals show the stereo camera pairs, red show the LIDARs in the front and back. The PCs are mounted on a rack behind the passenger seat. The picture also shows the ventilation system used to keep the setup cool.

I. Introduction

The DARPA Urban Challenge 2007 [1] was a milestone event in the area of intelligent transportation systems. It provided us with a glimpse of the not so distant future where autonomous unmanned vehicles can safely and reliably drive from one designated location to another in an urban environment with other manned and unmanned vehicle traffic. In order to accomplish its missions, these autonomous vehicles had to perform complex maneuvers such as merging, passing, parking and obeying intersection precedence, with no human intervention. Sarnoff Corporation participated in this event as part of Team Autonomous Solutions, which placed among the top twenty teams in the competition. We developed an integrated vision based sensing and positioning system for autonomous navigation composed of existing Sarnoff technologies and commercial off-the shelf (COTS) hardware.

While the field of autonomous mobile robotics has made a lot of progress in the recent past [2], truly deployable intelligent systems that work reliably using COTS components in dynamic everyday scenarios are seldom seen outside of laboratories. The 2005 DARPA Grand Challenge [3] saw some of the best robotics researchers successfully complete an off-road navigation competition with no traffic or dynamic obstacles. Extending this competition to the dynamic urban environment required significant technological advances in sensing systems and algorithms.

Similarly in the field of intelligent automotive systems, there has been a lot of work done in developing convenience and safety features based on both passive (vision) and active (radar, laser) sensing [4]. However at the time of the DARPA Urban Challenge there was no available integrated solution that could perform all the necessary sensing to enable autonomous driving in a complex urban environment. This motivated us to develop our integrated vision based sensing system for autonomous navigation which could estimate ego-motion, and detect and track road boundaries, in-path obstacles and stationary and moving vehicles. The key contribution of our work is a real-time primarily vision-based system that provides full spatial and temporal awareness of the environment.. Other systems [5], [6] have used single frame stereo vision and color based segmentation for obstacle detection and identifying easily traversable roads respectively. Those had no object identification (vehicle/non-vehicle) or temporal tracking and association strategies implemented which would enable the system to successfully navigate autonomously based primarily on the vision system. Several systems at the DARPA Urban Challenge used a high-definition Velodyne LIDAR [7], [8] that provided 1.8 million 3D points per second around the vehicle. While the Velodyne provides rich data, it is fairly expensive, conspicuous, and substantially changes a vehicle's profile. Our sensor suite is cheaper, more compact and can be mounted inside the vehicle, making it more attractive for both military and

Sponsored by the Defense Advanced Research Projects Agency (Urban Challenge program) under DARPA/CMO Contract No. HR0011-06-C-0151. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressly or implied, of the Defense Advanced Research Projects Agency or the U.S. Government. Distribution statement “A” (Approved for Public Release, Distribution Unlimited).

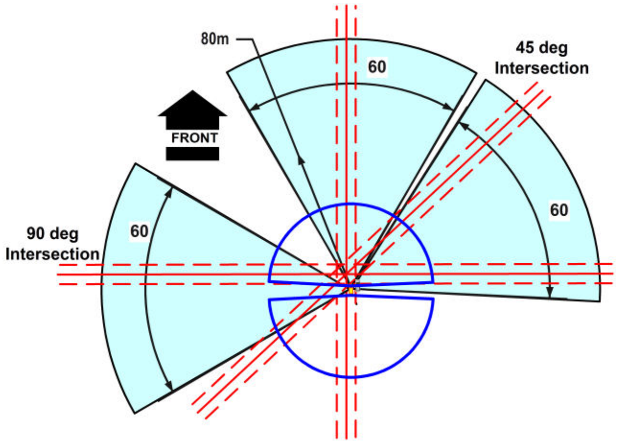


Fig. 2. Sensor Placement. This scale drawing illustrates the sensor horizons of the three stereo vision sensors (cyan sectors) and the front and back lasers (blue sectors) for the host vehicle (between the laser sectors in the image) standing on an intersection. Both perpendicular and 45 degree intersections are illustrated. The right stereo sensors had to be angled to satisfy short-range visibility requirement on intersections (refer to Fig.5). Approaching vehicles beyond 80m were specifically detected by the Motion Detection module.

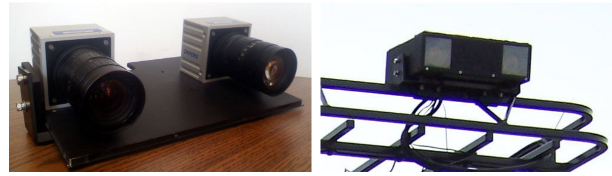


Fig. 3. Stereo Vision Sensor: Inside the rig (left) and mounted on the vehicle (right)

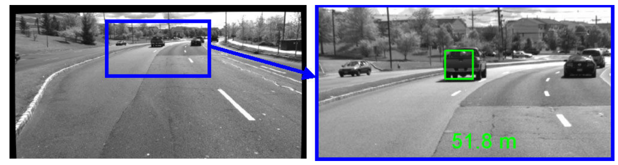


Fig. 4. WA (left) and NA (right) images. NA is a ROI of the original full resolution image. WA is the subsampled reduced resolution image. Notice how the narrower field-of-view of the NA image allows detection of a vehicle much farther out.

commercial autonomous robot users.

Section II describes the various system components and section III their integration. Lessons learned and future work are in Section V.

II. System Components

The smallest building block in our vision based system is a single high resolution digital monochrome camera used for optical flow based motion detection. We composed these cameras in pairs to perform real time dense stereo using a high-end image processor board. Next we synchronized and integrated each pair of cameras with an inexpensive inertial measurement unit (IMU), GPS and vehicle odometry (via the vehicle CAN bus) to implement a positioning system. Each of these integrated stereo modules was connected to a single computer and the communication between the computers was through ethernet. While a pair of static LADAR scanners were also mounted on the vehicle, these were used as independent sensors for redundant detection of static obstacles and not integrated with the vision system. We now present a discussion of the perception requirements for the Urban Challenge and the perception modules we developed to address them.

A. Perception Requirements

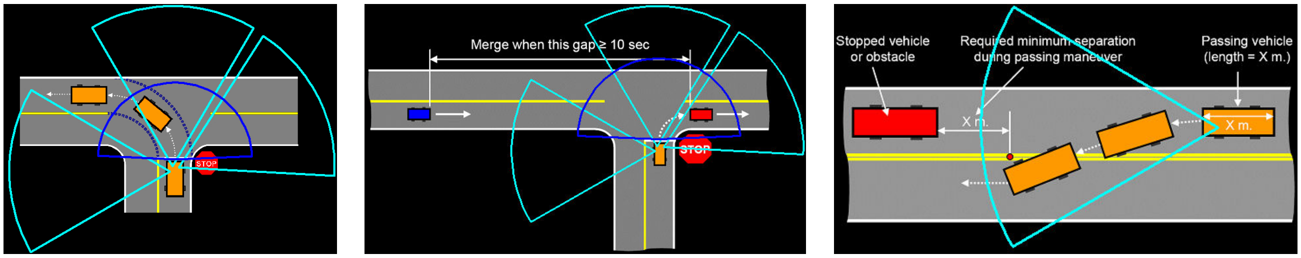
The choice of the imaging system was largely governed by the perception needs of an autonomous vehicle navigating through traffic and following traffic regulations. We performed experiments to identify the sensor horizon required in different directions on the basis of traffic maneuvers that needed to be performed. Assuming a maximum approach speed of 30 mph and a safety cushion of 10 seconds (following DARPA Urban Challenge restrictions), the approaching vehicles need to be detected at around 130 meters. In order to obtain sufficient image pixels on a vehicle at that distance while maintaining a wide-angle field of view, we decided upon three (pairs of) 60-deg. high-resolution (1920×1080) cameras (Imperx IPX-2M30H-L).

Each acquired high-resolution HD image (1920×1080) is smoothed by a gaussian filter (to avoid aliasing artifacts) and then subsampled to a resolution of 640×360 (henceforth called Wide Angle or WA). Since this image is a subsampled version of the acquired image, we get a full field-of-view sensing of 60 degrees at one-third the resolution. At the same time, we also extract a region-of-interest (ROI) of size 640×360 (henceforth called Narrow Angle or NA) from the acquired high-resolution image to capture one-third of the field-of-view (i.e. 20 degrees) at full resolution. Fig.4 illustrates how this design allows detection of both closer and farther obstacles with a single high-resolution camera. Dynamic selection of an appropriate ROI for the NA image can allow software panning of camera look-at direction. In our system we manually identified and fixed a suitable ROI for each NA image (independently for the front, left and right looking rigs). One stereo pair of cameras (stereo-rig) was used for each of the front, left and right viewing directions as depicted in Figs.1 and 2. This setup covered the visibility requirement for typical urban navigation scenarios (Fig.5).

The following subsections describe in detail the vision modules that were developed for the challenge.

B. Motion Detection

Moving traffic was an important component of the Urban Challenge and to address the problem of detecting moving objects specifically, we used a Motion Detection (MD) module. To detect moving objects from an image sequence, we need to compensate for host vehicle motion to nullify the apparent motion of the static environment. However, if the host vehicle is static, the motion compensation step can be skipped. For our real-time system, we employed MD whenever the host vehicle stopped thus addressing specific situations e.g., "stopped at an intersection".



(a) Left Turn (b) Right Turn (c) Passing

Fig. 5. Sensor Coverage. This drawing illustrates the angular coverage of the three stereo vision sensors (cyan sectors) and the front and back lasers (blue sectors) for the host vehicle executing a left-turn, a right-turn and a passing maneuver. Scenario schematics from the DARPA Urban Challenge guidelines.

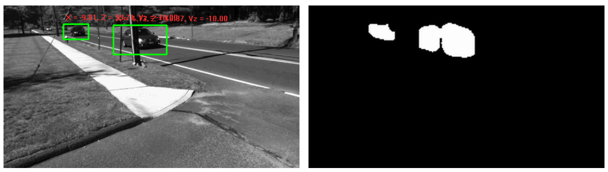


Fig. 6. Motion Detection. In this view from the left-camera, the algorithm computes a motion mask (right) from which the approaching vehicles are detected directly (left).

The motion detection algorithm is run independently on the left WA and the left NA image for each stereo pair. The algorithm works as follows:

1) Compute a dense optical flow using images at two successive timestamps and integrate over time.
2) On each frame, update a salience measure that is proportional to pixel motion in a consistent direction. For details, refer to [9], [10].
3) Filter the optical flow map in the salient regions to detect regions with a motion component toward the host vehicle.
4) Perform connected components analysis on the filtered motion field above, find bounding boxes corresponding to approaching vehicles and output the motion mask for use by the vehicle-detector (Fig.6).

We employed motion detection dynamically whenever the vehicle speed measure on the CAN bus was close to zero and remained so for a small finite amount of time. A temporal hysteresis was needed to ensure that the motion detection was not triggered on an isolated zero value (spike). Note that for the Urban Challenge, we assumed that the only objects approaching our vehicle were other vehicles. This worked quite well as a minimum blob size constraint proved enough to filter out other motion.

C. Stereo Processing

We rely on dense stereo for an obstacle detection module that reliably detects object boundaries and computes good distance estimates.

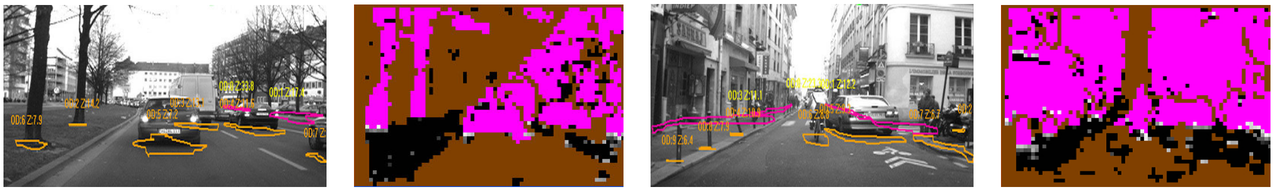
We perform stereo processing on a high-end video processing board - the Matrox Odyssey XPRO+. This board provides both high-level (Odyssey Native Library) and low-level (Odyssey Developer's Toolkit) APIs for writing custom image processing applications that can use an on-board Pixel-Accelerator ASIC, a customizable co-processor FPGA and a G4 PowerPC for control. We ported our real-time stereo processing algorithm [11] to this board. The current implementation gives us stereo output for the sub-sampled 640×360 (WA) images at 12 Hz without using the low-level API (which can lead to a 2× speedup in the best case). In practice, this proved to be sufficient for the host vehicle speeds required in the Urban Challenge.

Note that for the Urban Challenge, generic obstacle detection was required only for closer ranges because at longer ranges the system only needs to detect moving vehicles. Therefore, we only computed dense stereo for the WA images thus needing only one Matrox board per stereo pair.

D. Obstacle Detection

The stereo range-map at the lower resolution is used for obstacle detection upto 30m. The obstacle detection (OD) function (Fig.7) analyzes stereo derived range data together with vehicle attitude to associate a traversal cost with the 3D structure discovered by the stereo process. The traversal cost is in the form of a height (above a reference "ground" surface) cost and a slope (local change relative to reference surface) cost. The algorithm [12] compensates for ground resolution variation, pre-filters the range image to reduce noise and analyzes the data in XZ coordinates to create a grid map. This analysis uses a multi-resolution technique to estimate a function Y_{ref} that is used to compute the height cost map for each XZ location. The spread of heights at a single XZ location and differences in lowest elevation between neighboring pixels classifies each map point as an obstacle or not. Map pixels are aggregated to obtain cells of a constant size on the ground near the camera and used to estimate the slope cost.

The OD algorithm outputs a cost map (combining the height and slope cost maps above) in the image space which is a gray-level representation of the obstacle lethality (opposite of traversability) estimates. The obstacles in the scene are thus labeled with an intensity value of 255 while things close to be on the ground have a score value close to zero. The OD output for each stereo frame is converted to the vehicle-centric coordinate system and subsequently polygonized (around segmented obstacles) to overlay on the map grid. The frame by frame OD algorithm flow is captured below.



(a) (b) (c) (d)

Fig. 7. Obstacle Detection. Output of the obstacle detection algorithm showing the color coded cost maps (b),(d) using both height and slope thresholds for two typical urban scenarios. Also shown are the recovered obstacle polygons projected on the ground plane (a)(c). Notice the ability of the algorithm to recover individual obstacles with good boundaries even in the presence of a large amount of clutter.

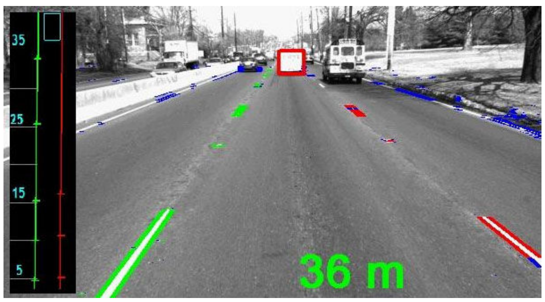


Fig. 8. Lane Detection. Green and red points are detected as the left and right lane-markers respectively. Blue points are the remaining non-lane features on the ground surface. Note how this detection of host lane boundaries enables selective detection of only vehicles in the host vehicle's lane.

1) Filter stereo range image (dense 3D data) and perform ground compensation.
2) Create an obstacle grid map (world XZ coordinates) of a chosen resolution with each cell having a height based cost and a (drivable) slope cost.
3) Based on thresholds on the combined height and slope costs generate a cost map image, assigning an obstacle lethality value (0-255) to each pixel and cluster 3D points identified as lethal obstacles.
4) Segment the clustered obstacle points after projecting the points on the ground XZ plane. Polygonize the resulting obstacle boundaries (Fig.14 shows OD polygon outputs).

E. Lane Detection and Tracking

In our system, lane detection (LD) was needed for two reasons. First, to aid the vehicle detector in making intelligent decisions about where to look and second, for autonomous navigation in the case of GPS loss or sparse waypoints by following lane boundaries. For both these purposes, we only need to detect lane-markings to a short range of about 30-35m and so we process only the WA images for this purpose.

Lane detection on the basis of image features only is prone to failure. In urban scenarios, due to the presence of a significant amount of clutter, spurious non-ground features can get detected as lane-markers if only image intensity information is used. To alleviate this problem, we make use of the depth information in conjunction with the image intensity information to guide our search. The algorithm [13] works as follows:

1) Recover 3D structure from the dense stereo available for WA images.
2) Identify 3D points that lie on the ground surface.
3) Fit a plane model to the ground surface.
4) Find lane markers on the detected ground surface using local contrast and edge-pairing methods.
5) Fit a linear model to the lane markers. On curved roads, fit a quadratic model of lane shape.
6) Track lane model with an Extended Kalman Filter using available vehicle speed and yaw-rate (see Fig.8).

F. Vehicle Detection and Tracking

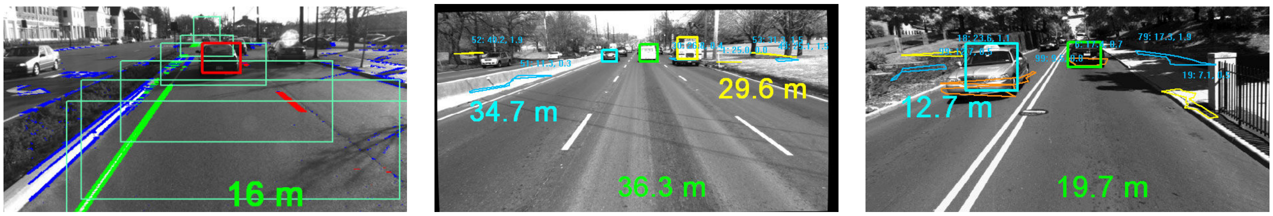
The detection of vehicles - both moving and static - is at the core of the Urban Challenge problem and to address scalability and real-time issues we developed a modular and fast algorithm to detect the presence of a vehicle in any specified image ROI. Specifically, we designed an algorithm to detect specifically the front or the back of a vehicle.

While driving down a road we need to detect vehicles that are directly ahead of us (to maintain proper gap or to pass them in case they are stalled) and vehicles in the oncoming lane (to avoid them before starting to pass). Also, on intersections we need to detect vehicles approaching from the front, left and right to properly merge into the traffic and also detect vehicles which have arrived before us on their stop lines (to do proper intersection precedence). Note that in all the above cases, it is important to distinguish vehicles from general obstacles as the behavior expected of the robot is different on encountering a vehicle than it would be when encountering some other obstacle. For example, in a parking lot or a free zone, it is ok to label all vehicles as obstacles as the expected behavior is to avoid these.

To handle any of the above situations, the vehicle detection (VD) algorithm needs to look only for the back or front of a vehicle. It is clear that the front cameras are always looking at the front or back of vehicles. The left and right cameras are only needed to handle the intersection scenarios and in these cases, even those are only looking at the front or back of vehicles (Figs.2 and 5).

Given an image ROI, the vehicle detection algorithm [13] works as follows:

1) Filter the input left and right images to locate vehicle like edge features.
2) Use the stereo depth map to compute a coarse scale of the expected vehicle size. In case a depth map is not



(a) (b) (c)

Fig. 9. Vehicle Detection. (a) Vehicle search ROIs selected on the basis of lane output, (b) output of VDs running on each of the three lanes (lane output not shown) and (c) example of a typical urban scenario with two-way traffic.

available, compute a coarse depth estimate by stereo matching on the edge features computed above[1].

3) Apply sub-pixel epipolar constraints (around the coarse scale determined above) to the detected features and filter out inliers consistent with a single disparity value.
4) Group these inlier features using a prior model on the 3D arrangement of such features for the front or back of a vehicle. If such a grouping is not possible, reject the input ROI as a possible vehicle candidate.
5) Compute a bounding box around the grouped features and apply a multi-layer adaboost classifier [14] to verify if it is a vehicle. If verified, output as a candidate vehicle box along with its distance estimate.

In order to achieve the required real-time performance, we resort to a "lane-search" approach where we apply the above algorithm to only specific discretized ROIs along a particular lane of the road in order of increasing distance. Once a vehicle is detected in a particular ROI, the search for another vehicle in that lane is abandoned. The location of these ROIs can come from the knowledge of the left and right lane-boundaries (if available e.g. Fig.9(a)) or can be fixed in the vehicle coordinate system by using the (calibrated) ground-plane to camera transformation and/or vehicle pitching information (from the Localization module). In the actual system, we used the latter approach whenever the lane-boundaries were not detected well. In addition to performance speedup, another advantage of the lane-search approach is that we can tune the VD to search for vehicles only in specific lanes. We will describe in section III how we adapt this to address the different urban challenge scenarios.

To further reduce false positives, we use the other available system components to support VD. In NA imagery, MD is a more reliable detector of approaching motion when the host vehicle is stopped and can help eliminate false vehicle detections from the left and right cameras. Similarly, in WA imagery, OD is a reliable detector of all the short-range obstacles (which include vehicles). Fig.10 depicts how we use the MD and OD modules to reject FPs from VD.

After detecting the candidate vehicle boxes in all the lanes, we track these vehicles across time using an Extended Kalman Filter on the distance and image location of these boxes. This provides us with range-rate (velocity) information which is critical to determining the safety envelope when merging into moving traffic.

[1]Note that this approach does not specifically depend on the dense stereo map and can therefore be used even for the NA image pairs (where we do not compute dense stereo).

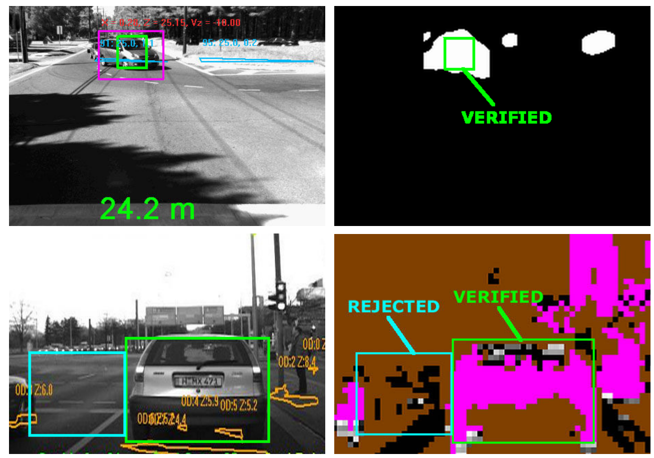


Fig. 10. Vehicle Detector FP removal. The motion mask from MD helps verify the detection from VD (top row). The cost mask from OD verifies the true detection from VD and correctly rejects the FP (bottom row). In each case, an analysis of the overlap of the detected ROI with the binary mask is used to verify or reject the detection.

G. Localization

The Sarnoff Video Inertial Navigation System (VINS) couples a GPS receiver, an IMU, and multiple cameras to perform localization in both GPS-available and GPS-denied areas. Detailed description of the VINS solution may be found in [15]. A brief overview is provided in this section. Central to the Sarnoff VINS solution is our Multi-camera Visual Odometry algorithm. This algorithm estimates camera pose (position and orientation) from image sequences and employs the following steps:

1) Detect and match feature points in each stereo pair; use epipolar and disparity constraints to eliminate false matches.
2) Compute 3D locations corresponding to these feature points using stereo triangulation.
3) Perform 2D-2D image feature matching over time to establish 3D-2D point correspondences.
4) Estimate camera pose using a robust resection method based on RANSAC followed by iterative refinement of the winning hypothesis.

The Visual Odometry algorithm only provides relative pose estimates. For absolute location and orientation in-

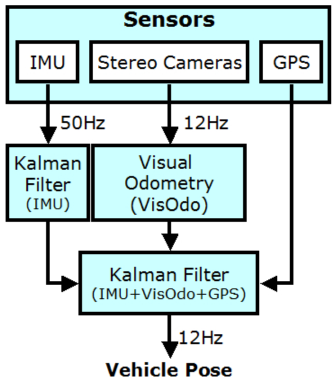


Fig. 11. VINS Block Diagram.

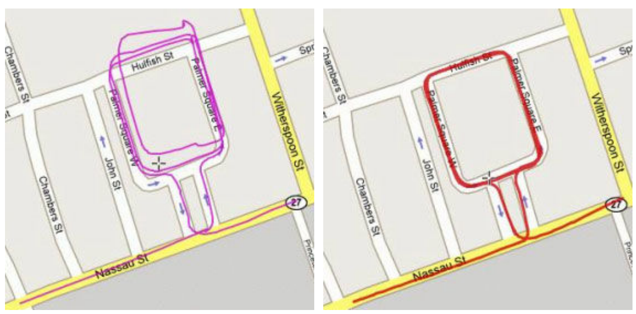


Fig. 12. Vehicle path output by a commercial system (left) that integrates a high-end GPS receiver with an IMU and the vehicle path recovered by VINS (right) during experiments conducted in Princeton, NJ. The main loop is around Palmer Square which has fully brick covered parts acting as an ‘urban canyon’.

formation, the Visual Odometry output is combined with GPS and IMU information as shown in Fig.11, VINS Block Diagram. Experiments (see for example Fig.12) have shown that even when GPS signals are completely denied, VINS localization drifts only 0.5-1% of the distance traveled.

III. System Integration

We will now describe how the different perception components were used together for robust environment perception by our robot. We divided the problem of perception into two different distance ranges - short range upto 40m and long range from 40 to 130m. These subranges correspond to the WA and NA imagery (described earlier) respectively. We will prefix the letters SR and LR to the name of a module to clarify whether it is running on a WA or a NA image (pair).

Fig.13 shows the data flow design of our system. Each WA image pair is processed by the stereo unit to compute a disparity map. This disparity map is utilized by the SRLD and OD modules to compute the lanes (L) and obstacles (O) respectively. At the same time, the SRMD module uses the left-WA image to compute a motion mask and then a set of approaching vehicles (V_a). Similarly, the LRMD module determines the approaching vehicles (V_a) using the left-NA image. For speed reasons, we do not run the stereo processor, LD or the OD modules on the NA imagery.

Multiple VD modules are run for each stereo pair. For the front camera, we run one VD for each lane on the WA images

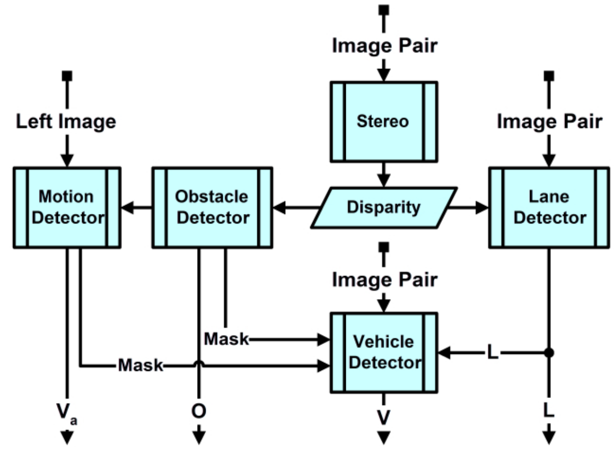


Fig. 13. Flow of data between modules. For simplicity, we have omitted the VINS module and also inputs like CAN data, IMU data etc.

TABLE I
Combination of modules under different scenarios

Scenario	Front Sensor Output	Left Sensor Output	Right Sensor Output
Driving	SRVD + OD $\rightarrow V$ OD $\rightarrow O$ SRLD $\rightarrow L$	Off	Off
Stopped at non-intersection	SRVD + OD $\rightarrow V$ SRMD + OD $\rightarrow V_a$ LRVD + LRMD $\rightarrow V_a$ OD $\rightarrow O$ SRLD $\rightarrow L$	Off	Off
Stopped at intersection	SRVD + OD $\rightarrow V$ SRMD + OD $\rightarrow V_a$ LRVD + LRMD $\rightarrow V_a$ OD $\rightarrow O$ SRLD $\rightarrow L$	Same as Front	Same as Front

thus detecting both vehicles ahead as well as oncoming upto 40m 9(c)(called the Short-Range Vehicle Detector SRVD). Similarly, we run one VD for each lane on the NA images for detection beyond 40m (called the Long-Range Vehicle Detector LRVD). Also, when the host vehicle comes to stop at an intersection, we switch the LRVD to look at an expanded lane of 50m and filter out only approaching traffic by using motion mask.

For the left and right cameras, we run a VD on the WA images only on the lane corresponding to oncoming traffic. This is because these cameras only help in the intersection scenarios where specific detection of oncoming traffic is more important (may be static or moving). For distance beyond 40m, since we are only interested in approaching traffic, we run one VD on an expanded lane of 20m (to take care of curved roads) and filter out only approaching traffic by using motion mask.

Table-I depicts the combination of different modules as outlined above, divided by scenarios. This illustrates how the different components helped make the system robust and fail-safe while alleviating the need for any single component

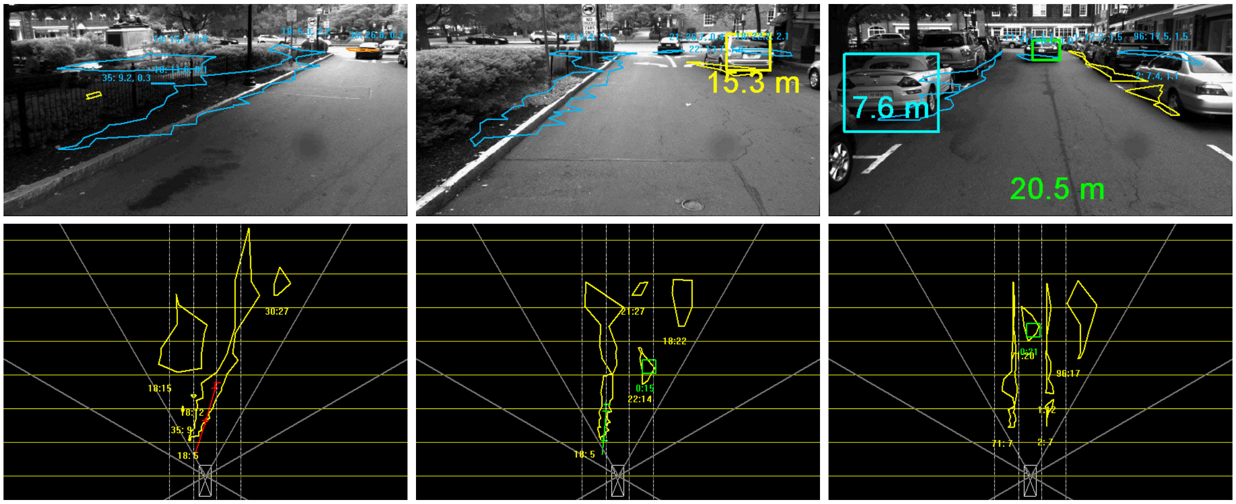


Fig. 14. System output for a cluttered urban scene. Vehicles and obstacles detected by the VD and OD modules respectively - image overlay (top row) and overhead map representation (bottom row). Yellow polygon: Obstacle, green box: Vehicle, green or red polyline: Lane-Boundary.

to handle all situations all by itself.

IV. Lessons Learned and Future Work

Our system was used to drive an autonomous vehicle during the DARPA Urban Challenge, showing that passive sensing is capable of providing reliable perception of the environment, especially for obstacle detection and tracking, and self-localization. Among the key lessons learned during the development and testing phases was that close co-operation between the vehicle planning and perception modules can compensate for small azimuth errors ($\sim$1m) in sensing, but not for false positives in the path of the vehicle. For this reason, we expended considerable effort to use multiple cues (motion, appearance and depth) to eliminate false positives ahead of the vehicle out to 30m. System testing and operation also benefited considerably from freezing of perception software development at least one month prior to the Challenge and working around bugs and missing features.

Future work will focus on more typical urban scenarios. These present situations such as narrow lanes without dividers, vehicles densely parked along curbs, unexpected appearance of vehicles from hidden driveways and unpredictable movement of people or animals. For path-planning under limited GPS (due to urban canyons from tall buildings), information about unknown fixed infrastructure, e.g., buildings, road intersections, stop lines/signs, could be very important since these provide information about potential paths to take, or locations of traversable free space. In dynamic and cluttered environments, the perception system needs to not just detect obstacles in and around the robot's path, but also recognize whether the obstacle is a vehicle, a truck, a bicycle, or a pedestrian, and whether the obstacles are moving or stationary. In these scenarios, we believe that vision-based perception, in combination with other sensors like LIDAR and Radar, can exploit important visual cues (e.g. brake light or turn signal of the vehicle ahead) and provide a more complete representation of the environment.

V. Acknowledgment

The authors would like to thank the members of Team Autonomous Solutions (Autonomous Solutions Inc and DeVivo AST) for their contributions during system development and testing. We would also like to thank our colleagues at Sarnoff Corporation for their help and support at various stages of the project.

References

[1] "The DARPA Urban Challenge." [Online]. Available: http://www.darpa.mil/GRANDCHALLENGE/

[2] R. Siegwart and I. R. Nourbakhsh, *Introduction to autonomous mobile robots*. Cambridge, MA: MIT Press, 2004.

[3] *The 2005 DARPA Grand Challenge: The Great Robot Race*, ser. Springer Tracts in Advanced Robotics. Berlin / Heidelberg: Springer, October 2007, vol. 36.

[4] "Volvo Collision Warning with Auto Brake." [Online]. Available: http://www.mobileye.com/uploaded/PressPDF/Volvo.AutoBrake.pdf

[5] A. Broggi, C. Caraffi, P. Porta, and P. Zani, "The Single Frame Stereo Vision System for Reliable Obstacle Detection Used during the 2005 DARPA Grand Challenge on TerraMax," *Intelligent Transportation Systems Conference, 2006. ITSC'06. IEEE*, pp. 745–752, 2006.

[6] Sebastian Thrun and others, "Stanley: The robot that won the DARPA Grand Challenge: Research Articles," *J. Robot. Syst.*, vol. 23, no. 9, pp. 661–692, 2006.

[7] "Team MIT Urban Challenge Technical Report." [Online]. Available: http://hdl.handle.net/1721.1/39822

[8] "Team Tartan Racing Technical Paper." [Online]. Available: http://www.darpa.mil/GRANDCHALLENGE/TechPapers/Tartan_Racing.pdf

[9] L. Wixson, "Detecting Salient Motion by Accumulating Directionally-Consistent Flow," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 774–780, 2000.

[10] S. Das and M. Aggrawal, "Multiple object tracking using probabilistic models," White Paper, Sarnoff Corporation, May 2005.

[11] G. van der Wal, M. Hansen, and M. Piacentino, "The Acadia vision processor," *Computer Architectures for Machine Perception, 2000. Proceedings. Fifth IEEE International Workshop on*, pp. 31–40, 2000.

[12] A. Stentz, A. Kelly, P. Rander, H. Herman, O. Amidi, R. Mandelbaum, G. Salgian, and J. Pedersen, "Real-Time, Multi-Perspective Perception for Unmanned Ground Vehicles," *Proc. of AUVSI Unmanned Systems Symposium*, vol. 2003, 2003.

[13] B. Southall *et al.*, private communication, Sarnoff Corporation, 2005.

[14] P. Viola and M. Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features," *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition*, 2001.

[15] D. Nister, O. Naroditsky, and J. Bergen, "Visual odometry for ground vehicle applications," *Journal of Field Robotics*, vol. 23, 2006.