2014 IEEE International Conference on Robotics & Automation (ICRA)
Hong Kong Convention and Exhibition Center
May 31 - June 7, 2014. Hong Kong, China

A Multi-Sensor Fusion System for Moving Object Detection and Tracking in Urban Driving Environments

Hyunggi Cho, Young-Woo Seo, B.V.K. Vijaya Kumar, and Ragunathan (Raj) Rajkumar

***Abstract*— A self-driving car, to be deployed in real-world driving environments, must be capable of reliably detecting and effectively tracking of nearby moving objects. This paper presents our new, moving object detection and tracking system that extends and improves our earlier system used for the 2007 DARPA Urban Challenge. We revised our earlier motion and observation models for active sensors (i.e., radars and LIDARs) and introduced a vision sensor. In the new system, the vision module detects pedestrians, bicyclists, and vehicles to generate corresponding vision targets. Our system utilizes this visual recognition information to improve a tracking model selection, data association, and movement classification of our earlier system. Through the test using the data log of actual driving, we demonstrate the improvement and performance gain of our new tracking system.**

I. INTRODUCTION

The 2005 DARPA Grand Challenge and the 2007 Urban Challenge offered researchers with unique opportunities to demonstrate the state of the art in the autonomous driving technologies. These events were milestones in that they provided opportunities of reevaluating the status of the relevant technologies and of regaining the public attention on self-driving car development. Since then, the related technologies have been drastically advanced. Industry and academia have reported notable achievements including: autonomous driving more than 300,000 miles in daily driving contexts [19], intercontinental autonomous driving [3], a self-driving car with a stock-car appearance [20], and many more. Such developments and demonstrations increased possibility of self-driving cars in near future.

After the Urban Challenge, Carnegie Mellon University started a new effort to advance the findings of the Urban Challenge and developed a new autonomous vehicle [20] to fill the gap between the experimental robotic vehicles and consumer cars. Among these efforts, this paper details our perception system, particularly, a new moving objects detection and tracking system. The Urban Challenge was held in a simplified, urban driving setup where restricted vehicle interactions occurred and no pedestrians, bicyclists, motorcyclists, traffic lights, GPS dropouts appeared. However, as shown in Figure 1, to be deployed in real-world driving environments, autonomous driving vehicles must be capable of safely interacting with nearby pedestrians and vehicles. The prerequisite to safe interactions with nearby objects is reliable detection and tracking of moving objects.

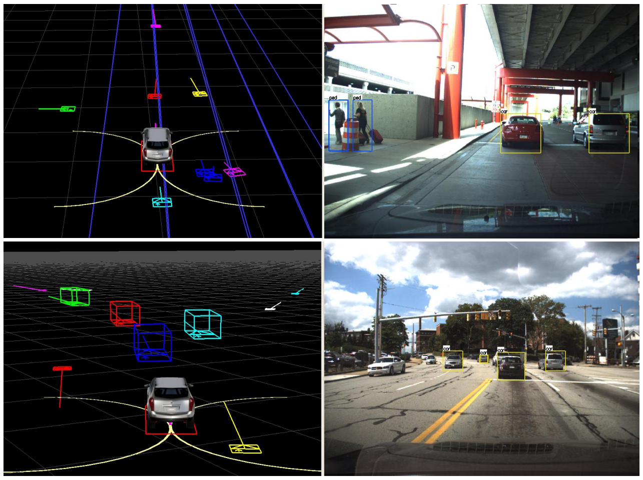

Fig. 1. Sample images show urban driving environments and screen-captures of our tracking system's results. The images in the first row show detection and tracking results from an arriving area of Pittsburgh international airport. The other two images in the second row show those of an urban street.

To develop such a reliable perception capability for autonomous urban driving, we redesigned our sensing system, extended our earlier moving obstacle tracking system and introduced new sensors in different modalities. Section III and Section IV detail the configuration of multiple sensors in different modalities. Knowledge of moving objects' classes (e.g., car, pedestrian, bicyclists, etc.) is greatly helpful to reliably track them and derive a better inference about driving contexts. To acquire such a knowledge, we exploit vision sensors to identify the classes of moving objects and to enhance measurements from automotive-grade active sensors, such as LIDARs and radars. Section V describes interactions between our vision sensor based object detection system and active sensor based object tracking system. Section VI discusses the experimental results and the findings. Section VII summarizes our work and discusses future work.

II. RELATED WORK

Detection and tracking of moving objects is a core task in mobile robotics and as well as in the field of intelligent vehicles. Due to such a critical role, this subject has been extensively studied for the past decades. Since a comprehensive literature survey of this topic is beyond the scope of this paper (we refer to [12], [16] for such surveys), here we review only the earlier work on multi-sensor fusion for

H. Cho, B.V.K Vijaya Kumar, and Ragunathan (Raj) Rajkumar are with the ECE Department and Young-Woo Seo is with the Robotics Institute, Carnegie Mellon University, 5000, Forbes Ave., Pittsburgh, PA 15213, USA. {hyunggic, kumar, raj}@ece.cmu.edu, young-woo.seo@ri.cmu.edu

978-1-4799-3685-4/14/$31.00 ©2014 IEEE

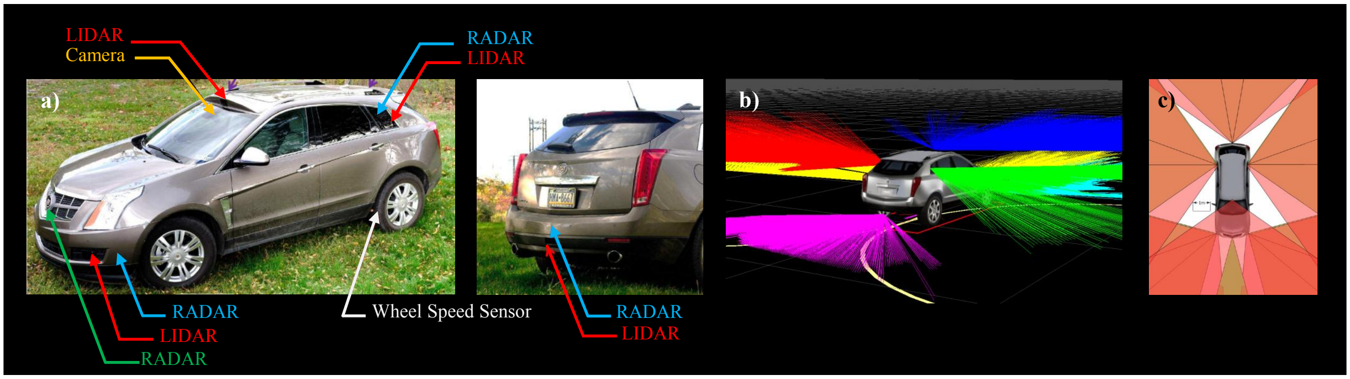

Fig. 2. CMU's new autonomous vehicle and its sensor configuration. (a) CMU's new autonomous vehicle is designed to minimize alterations of a stock-car appearance while installing multiple sensors to maximize the sensing coverage. (b) Visualization of LIDAR measurement. LIDAR scans acquired from individual sensors are depicted in different color. (c) A horizontal field of view (HFOV) of sensing coverage, emphasizing the coverage around the vehicle.

moving object detection and tracking, which are relevant to our work.

The Navlab group at Carnegie Mellon University has a long history of development of autonomous vehicles and advanced driver assistance systems. For the Navlab 11, one of the latest developments, they proposed a high-level fusion approach for object tracking using cameras and LIDARs [1]. In fact, our feature extraction algorithm for LIDAR is motivated by their method [12]. Another interesting effort was the work of Stiller et al. [18], where they used radar, LIDAR, and stereo vision for obstacle detection and tracking. Although they did not provide quantitative results of the system, it brought researchers' attention to the multi-sensor fusion approach.

Since then, an approach of fusing LIDAR measurements with vision sensors' outputs has gained popularity for vehicle tracking [11], [14] and pedestrian tracking [16], [17]. Monteiro et al. [14] used a single-layer LIDAR and a monocular camera to detect, track, and classify objects. For the fast detection and tracking, a LIDAR was used and generated regions of interest (ROIs) to a vision module. For the classification of objects, two classifiers, a Gaussian Mixture Model (GMM) classifier for a LIDAR and an Adaboost classifier for a camera, are applied. A sum decision rule was used to combine both outputs. Mählisch et al. [11] focused on a 'cross-calibration' method between these two sensors while showing vehicle tracking. Premebida et al. [16] used a multi-layer LIDAR and a monocular camera for pedestrian detection. They exploited several features for each sensor measurements and classification algorithms for better accuracy. Spinello and Siegwart [17] also utilized a multi-layer LIDAR for detecting hypotheses for pedestrians and then a vision-based pedestrian detector was applied for verification. A Bayesian decomposed expression was used as a reasoning fusion rule.

TABLE I
INSTALLED SENSORS SPECIFICATIONS AND THEIR PRIMARY USAGES.

Sensor Type	HFOV (°)	MaxRange (m) /Resolution	Update Rate (Hz)	Tracking Features
LIDAR	85∼110	200	50	edge target
RADAR1	30	250	12.5	point target
RADAR2	20 (near) 18 (far)	60 (near) 175 (far)	20	point target
Video Camera	45	640×480	8	vision target
FLIR Camera	36	640×480	24	vision target

The 2007 DARPA Urban Challenge provided researchers with a unique opportunity to develop and test the multi-sensor based systems [10], [15], [13]. Due to the practical nature of the competition, high-level fusion approaches were widely exploited. In particular, the Stanford [15] and MIT [10] teams developed a similar object tracking system which utilized a set of LIDAR sensors as primary sensors. Their systems first removed irrelevant measurements, such as laser scans from the ground and from vertical structures, and then fitted geometric primitives (e.g., 2D rectangles) to the remaining measurements to, using Bayesian filters, estimate objects' position, velocity, and size. Similarly, the Cornell team [13] used a Rao-Blackwellized particle filter for moving object tracking, where a data association problem is solved by a particle filter and a state estimation problem is solved by an extended Kalman filter. Most of the teams developed their own tracking systems to effectively fuse sensor measurements in different modalities. However, due to reliability issue and computational cost, a tracking system based on vision-sensor was not extensively studied for the competition.

III. A MULTI-SENSOR SETUP FOR ROBUST PERCEPTION

The underlying ideas of our sensor configuration are to 1) minimize any potential alterations of a vehicle's original appearance, 2) completely cover the area around the vehicle within a certain range, and 3) utilize existing, off-the-shelf and stock-car grade sensors. Based on these guidelines and prior experiences, we built a new sensing system as shown [20] in Figure 2(a). All sensors are seamlessly integrated into the vehicle chassis and their appearance is indistinguishable from outside. In particular, we installed six radars, six LIDARs, and three cameras. A radar is paired with a LIDAR at different heights. We did this to maximize the reliability and range of measurements. With our current sensor layout, any objects within 200 meters will be projected onto our sensing coverage and any objects within 60 meters or so will be seen by at least two different types of sensors (i.e., radar and LIDAR, or radar and camera). Figure 2(b) illustrates measurements from all six LIDAR sensors. For the vision sensors' setup, a camera is installed, in a forward-looking manner, inside the front window, next to the rear-view mirror and another is installed at the rear bumper to provide the front and back side of perspective images. The third camera is a thermal camera that captures scenes in infrared spectrum to perceive objects in challenging driving conditions, such as at night and in fog. Table I details the types and specifications of our sensors. All these sensors are stock-car grade and readily available on the market. Figure 2(c) depicts the blind spots. Due to the integration of multiple wide-FOV sensors the blind spots are small enough that no vehicle will be overlooked.

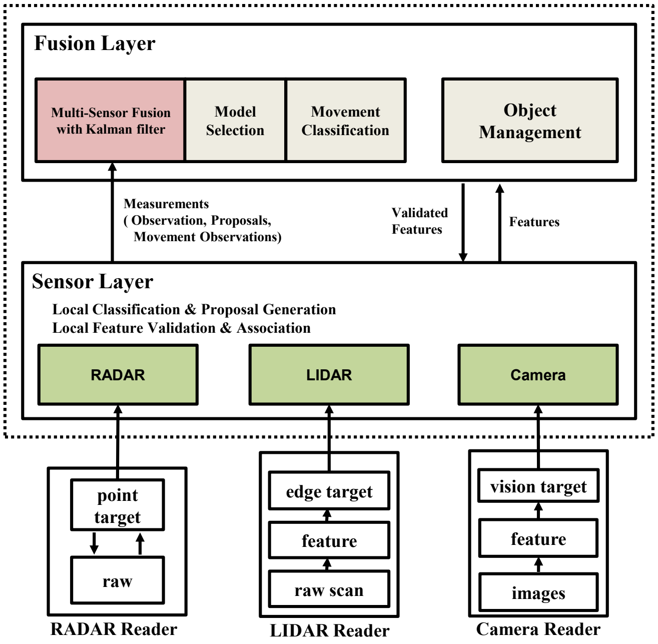

Fig. 3. A diagram of our tracking system. Our system is mainly comprised of two layers: a sensor and a fusion layer. We enhance and improve the architecture of our earlier tracking system [6].

IV. SENSOR CHARACTERIZATION

It is a challenging to seamlessly fuse measurements from 14 sensors and to generate tracking results consistent over time. To effectively address such a challenge, we extended, based on lessons learned from participation of several autonomous vehicle competitions, our earlier tracking system [6] and introduced new methods to effectively tackle real-world perception problems occurring in urban autonomous driving. Figure 3 shows a diagram that describes our new tracking system. Our system consists of two parts: sensor and fusion layer. By taking care of hardware specific operations, the sensor layer offers a separation between actual sensing hardware and specific tasks regarding detection and tracking of objects. By this way, the tasks at the fusion layer can be developed without knowing the details about the lower-level's sensing mechanisms. Each sensor reader acquires raw sensor data and extracts features (e.g., lines or corners), if any, and publishes them in a shared communication channel. A task at a higher-level, fusion layer, can pick up these features from the channel for its purpose. For example, based on lower-level's features, e.g., point or polygonal shapes, we execute point or box models to track the feature over time. For the underlying rationale of such a tracking architecture, we refer readers to [6].

Once measurements from any sensors are delivered to the fusion layer, they are treated similarly as units of measurements, but represented differently based on their sensing modalities. For example, a radar provides 2-dimensional position and velocity of an object. It usually reaches objects relatively farther distance (e.g., more than 200 meters) from a host vehicle and offers a direct velocity measurement. We represent radar measurements at time step k as

$$\mathbf{z}^R(k) = \{\mathbf{r}_1, \mathbf{r}_2, ..., \mathbf{r}_p\} \tag{1}$$
$$\mathbf{r}_i = [x \; y \; \dot{x} \; \dot{y}]^T \quad i = 1, ..., p$$

where $\mathbf{r}_i$ is a point position and velocity measurement with respect to the radar sensor coordinate and p is the number of radar measurements at time step k.

By contrast, measurements from LIDAR sensors provide 3-dimensional point clouds. Mostly these point measurements are dense enough to partially or completely delineate the shape of objects. Note that the actual formation of point clouds and their coverages of objects' shapes are dependent upon various factors, e.g., field of view (FOV), angular resolutions, line of sight between a sensor and an object. A high-density measurement comes with additional processing cost because it is necessary to pre-process (e.g., segmentation or features, like line segments or corners, extraction) point clouds to make them attached to objects to track. For example, to keep tracking the vehicle right front of a car, one needs to know which of point clouds are parts of the vehicle (i.e., segmentation) and to represent that clustered point cloud as a computational form (i.e., feature extraction – represent the clustered point as a line).

For representing LIDAR measurements, we treat six, four-plane LIDARs as one homogeneous sensor, analyze their measurements using built-in segmentation and extract features, like line segments or junctions of lines ("L") shape [12]. LIDAR measurements at time step k are expressed by:

$$\mathbf{z}^L(k) = \{\mathbf{l}_1, \mathbf{l}_2, ..., \mathbf{l}_q\} \tag{2}$$
$$\mathbf{l}_i = [x \; y \; \phi \; \dot{x} \; \dot{y} \; w \; l]^T \quad i = 1, ..., q$$

where $\mathbf{l}_i$ consists of the position of the center of the box (fitted by the feature), orientation (ϕ), velocity, width (w), and length (l) of the box. In fact, w is computed as $\max(OW, e1)$, where OW is the canonical width of that

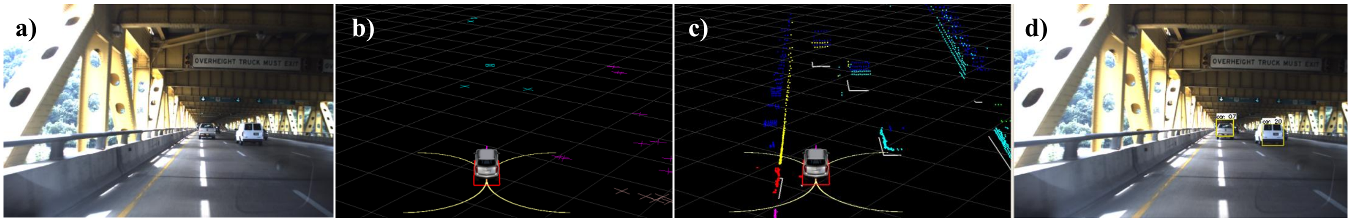

Fig. 4. Example images show raw sensor data and refined raw data as measurements. (a) An input scene. (b) Raw data from six radars. The data are used as features (called 'point target') for tracking directly. (c) Raw scan data from six LIDARs. "L" shaped features (called 'edge target') are extracted for tracking. (d) Bounding boxes from the vehicle detection system are used as features (called 'vision target') for tracking.

object class and $e1$ is the actual measured length of a short edge of the feature. The same idea applies to l. q is the number of LIDAR measurements at time step k.

Lastly, cameras provide high-definition snapshots of scenes. While rich information in the image frames makes vision data interesting, determining what features to extract and how to interpret them for detection and tracking of moving (or even static) objects is still an active research topic. To effectively utilize visual information, we developed a vision-based object detector that aims at identifying pedestrians, bicyclists, and vehicles [5], [4]. For sensor fusion purpose, we represent the detected objects using bounding boxes and treat them as measurements from vision sensors. Then camera measurements at time step k is expressed by:

$$\begin{aligned} \mathbf{z}^C(k) &= [\mathbf{c}_1, \mathbf{c}_2, ..., \mathbf{c}_r] \\ \mathbf{c}_i &= [x1\ y1\ x2\ y2\ class]^T \quad i = 1, ..., r \end{aligned} \tag{3}$$

where ($x1$, $y1$) and ($x2$, $y2$) are the coordinates of the left-top and right-bottom point of a bounding box in the image space, respectively and $class$ indicates an object class (i.e., pedestrian, bicyclist, or vehicle). r is the number of bounding box measurements at time step k.

In summary, we represent measurements from different sensors differently based on individual sensors' acquisition principles and operating characteristics, but treat them the same way to facilitate the information fusion process. Our tracking system takes measurements from three different types sensors at time step k as:

$$\mathbf{z}(k) = \{\mathbf{z}^R(k),\ \mathbf{z}^L(k),\ \mathbf{z}^C(k)\} \tag{4}$$

In practice, these measurements are asynchronously acquired by each sensor and are timestamped to be published on the data communication channel. Figure 4 shows an example of those sensor data and extracted features.

V. MULTI-SENSOR FUSION

This section details how to fuse sensor's measurements to accurately detect and consistently track neighboring objects. For the tracking, we implemented an Extended Kalman Filter (EKF). To effectively apply this filter to our setup, we employ the sequential-sensor method [7] that treats observations from individual sensors independently and sequentially feeds them to the EKF's estimation process. We choose such a method to sequentially process multiple, heterogeneous measurements arriving in an asynchronous order.

A. *Tracking Models*

This work aims at developing a tracker that, using multiple sensors in different modalities, reliably tracking pedestrians, bicyclists, and vehicles. To effectively handle the constraints and characteristics of target objects' motions, we use two motion models: a point model ($\mathcal{M}_P$) and a 3D box model ($\mathcal{M}_B$). In particular, we expanded our earlier, 2-dimensional box model [6] to 3-dimensional one, to realistically represent the detected objects. For the three different sensing modalities, we devise three observation models: Radar ($\mathcal{O}_R$), LIDAR ($\mathcal{O}_L$), and camera ($\mathcal{O}_C$) observation model.

$$\begin{aligned} \mathcal{M} &: \{\mathcal{M}_P, \mathcal{M}_B\} \\ \mathcal{O} &: \{\mathcal{O}_R, \mathcal{O}_L, \mathcal{O}_C\} \end{aligned} \tag{5}$$

Motion Models: Each of three moving objects of interest (i.e., pedestrians, bicyclists, and cars) has its own motion kinematics and constraints. For example, a pedestrian can move in any directions whereas the motions of a vehicle or a bicyclist is confined by non-holonomic constraints. To estimate these motions, we use two motion models: a point model and a 3D box model. For the point model, we assume an object moves with a constant acceleration [2]. We represent the state of the moving point at time step k by its 2-dimensional coordinates, velocities, and accelerations:

$$\mathbf{x}(k) = [x(k)\ y(k)\ \dot{x}(k)\ \dot{y}(k)\ \ddot{x}(k)\ \ddot{y}(k)]^T \tag{6}$$

Our 3D box model is a bicycle model [9] with its estimated 3D cuboid. The state of a 3D box model is represented by

$$\mathbf{x}(k) = [x(k)\ y(k)\ \phi(k)\ v(k)\ \omega(k)\ a(k)\ w(k)\ l(k)\ h(k)]^T \tag{7}$$

where (x, y), ϕ, v, ω, and a are the position of the center of the box, yaw angle, velocity, yaw rate, and acceleration. The yaw angle defines the orientation of the velocity and acceleration vectors. The volume of a 3D box is defined by its components, width, w, length, l, and height, h.

Observation Models: We devise three different observation models for each of three different sensing modalities: $\mathcal{O}_R$, $\mathcal{O}_L$, and $\mathcal{O}_C$.

The radar observation model ($\mathcal{O}_R$) aims at modeling observations about a point target. It is designed to process direct position and velocity measurements.

The LIDAR observation model ($\mathcal{O}_L$) is primarily used to model a box target. This is a nonlinear mapping of the state

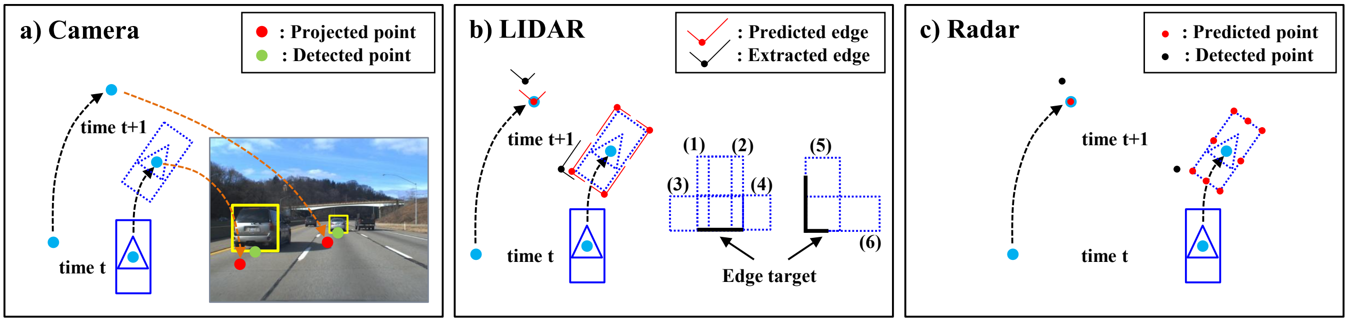

Fig. 5. Illustration of data association methods for each sensor. (a) Camera: predicted moving object hypotheses are projected into the image space and then associated with a set of detected ‘vision targets’. (b) LIDAR: a set of possible ‘edge targets’ are generated from the predicted moving object hypotheses and then associated with a set of extracted ‘edge targets’. (c) Radar: a set of possible ‘point targets’ are generated from the predicted moving object hypotheses and then associated with a set of detected ‘point targets’.

space into the LIDAR’s measurement space.

$$\begin{bmatrix} x \\ y \\ \phi \\ \dot{x} \\ \dot{y} \\ w \\ l \end{bmatrix} = \begin{bmatrix} x(k) \\ y(k) \\ \phi(k) \\ v(k)\cos(\phi(k)) \\ v(k)\sin(\phi(k)) \\ w(k) \\ l(k) \end{bmatrix} + \mathbf{v}(k) \tag{8}$$

where $\mathbf{v}(k)$ is the measurement noise at time step k. To make this noise realistic, one needs to analyze a collected, labeled LIDAR data set to derive its statistics such as covariance matrix. Our LIDAR observation model $\mathcal{O}_L$ is derived to support both the point motion model $\mathcal{M}_P$ and the 3D box model $\mathcal{M}_B$. For example, when $\mathcal{O}_L$ is used for $\mathcal{M}_P$, only the position measurement is used to update the state, where the position corresponds to the center of the edge that is closer to the host vehicle.

The last observation in our system is the camera observation model ($\mathcal{O}_C$). The camera observation model is primarily used to deal with bounding box measurements in the image plane. However, due to depth ambiguity, we do not use such bounding box detections to update motion estimation, but use the detection results to estimate the width and the height of an object and determines objects’ classes. Accordingly, $\mathcal{O}_C$ cannot be used for a new object initialization or termination. If the data association between image frames is correctly done, it is straightforward to compute the relationship between a pixel height $y2 - y1$ (width $x2 - x1$) and a physical height $h(k)$ (width $w(k)$), based on the camera geometry: $y2 - y1 \approx h(k)f_p/d$, where f_p is the focal length expressed in pixels and d is a distance which we can estimate in a precise manner via radars and LIDARs. Based on this, we define the camera observation model ($\mathcal{O}_C$) for a box model ($\mathcal{M}_B$) as

$$\begin{bmatrix} x2 - x1 \\ y2 - y1 \end{bmatrix} = \begin{bmatrix} w(k)f_p/d \\ h(k)f_p/d \end{bmatrix} + \mathbf{v}(k) \tag{9}$$

Note that the $\mathcal{O}_C$ can support only the $\mathcal{M}_B$ since a $\mathcal{M}_P$ model does not have the concept of shape.

B. Data Association

It is critical to associate the current measurements with the earlier state variables, to optimally estimate the state of tracking objects. This section details the improvement we made on, by utilizing our new camera observation model, our previous data association algorithm [6] for radar and LIDAR sensors.

Firstly, to associate camera observations (we call vision targets) over frames, we project the center of the predicted moving object hypotheses, represented by either a point or a box model, onto the next image frame under the pinhole camera model. After the projection, we search for the nearest neighbor that minimizes the distance between the projected point and the mid of the bottom line of the detected bounding boxes. Figure 5(a) illustrates this search. Once such an association is successfully made, the camera observation and its object classification is instantiated using equation 9. For the box model, its volume is also associated as well as its object class membership. For a point model, the observation is instantiated only with its class membership.

Secondly, for the association of LIDAR observations (we call edge targets), we generate, based on the predicted moving object hypotheses, a set of possible alignments of edge targets. There are four alignments for a box model and one for each point model. The left side of Figure 5 (b) illustrates such an alignment generation. Similar to the camera measurement association, the extracted edge targets are associated to the closest predicted one that minimizes the distance of the corner points. If an extracted edge target is associated to a predicted box model, all possible interpretations of the edge target as the box model are generated as illustrated in the right side of Figure 5 (b). Among the interpretations, one that has the maximum overlap with the predicted box is chosen to generate the observation. In practice, however, we found that considering all possible interpretations of an edge target occasionally fails to correctly match edge targets.

To improve our earlier association method, we utilize the vision target. For example, when our vision object detector returns a highest response of a vehicle’s rear view, we ignore irrelevant alignment (e.g., side-view alignments of edge targets). For example, in Figure 5(b), the alignments, (3), (4), and (6) are hypotheses about a vehicle's side-view and hence are ignored when the vision target casts a vote for a vehicle's rear-view.

Finally, for data association of radar observations (we call point targets), a set of possible point targets is generated from the predicted moving object hypotheses. Since radars are usually poor in determining a lateral position of an object, when a tracked object is modeled as a 3D box model, we generate multiple points along the contour of the box model. If an object is tracked through a point model, we generate a single point. The association between the predicted and the actual measurement is made by the nearest-neighbor approach. Figure 5(c) illustrates such a radar measurement association.

C. *Movement Classification*

Knowing whether a detected object has non-zero motion is important to optimally estimate the state of a tracking object. This is particularly true for urban driving scenarios where there is frequent stop-and-go traffic, queuing at traffic signals, abnormal vehicle interactions, and so forth. In principle, a tracking system should be able to trace trajectories of any moving objects around the host-vehicle. However, it is challenging to reliably track an object that was moving and now temporarily stops, but is going to move in the near future. To track such irregular temporal patterns, it is necessary to keep a record about series of motions as well as being determining whether an object is in motion. To implement this idea, our previous system [6] introduced two movement flags about 1) the movement history, i.e., *observed moving* and *not observed moving* and 2) the movement state, i.e., *moving* and *not moving*. The flag *moving* is set when the tracking system decides the object is currently in motion. The flag *observed moving* is set when the tracking system determines that the position of a tracked object has significantly changed. For the classification of the current movement state, the direct movement observations from radars was used. Since LIDARs do not provide a direct movement confirmation, the statistical test which compares an objects estimated velocity with a threshold v_{min} was used. For the classification of the movement history state, the distance traveled is computed from the last time stamp that the object has been classified as *not observed moving*. Then this distance is compared with a threshold, $d_{traveled}$. In practice, it is very hard to set up a single set of parameters that works well for different object class. For example, parameters optimized for vehicles do not work well for pedestrians. Thus, during the development phase, we empirically found multiple sets of parameters that work optimally for each object class.

VI. EXPERIMENTS

To evaluate the performance of our new multi-sensor, object tracking system, we drove our robotic vehicle and collected data (i.e., images, radar points, and LIDAR scans) in about a 25-minute driving. The route is comprised of a mix of streets and inter-city highways between Carnegie Mellon University's campus and Pittsburgh international airport. The distance is about 20miles. We first describe the system setup for the detection and tracking system and then discuss the evaluation results.

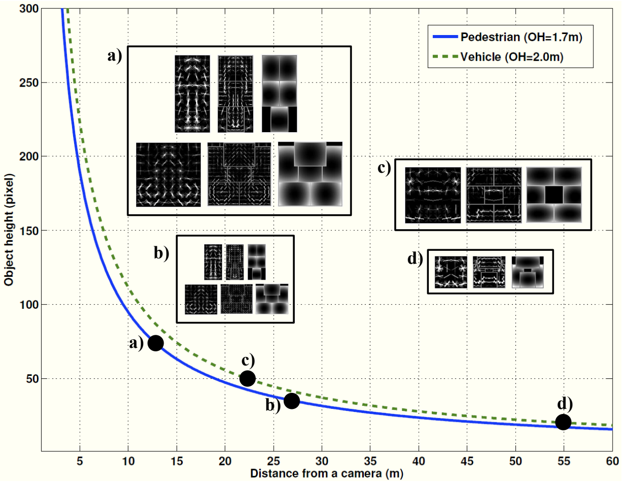

Fig. 6. Pixel height in image space as a function of distance d from the camera. Based on this analysis, all models are designed and visualized here. (a) Normal-sized pedestrian/bicyclist model (72×32 with 8×8 HOG cell). (b) Small-sized pedestrian/bicyclist model (36×16 with 4×4 HOG cell). (c) Normal-sized vehicle model (48×48 with 8×8 HOG cell). (d) Small-sized vehicle model (16×16 with 4×4 HOG cell).

A. *Experimental Setup*

Our tracking system runs on a computing cluster that consists of four mini-ITX, form-factor computers (i.e., Core 2 Extreme QX9300@2.53GHz, 8GB RAM). Each of the sensors generates its measurements at its own operating cycle (See Table I). Software modules read measurements from individual sensors and publish them through an inter-process communication channel. While doing so, measurements acquired at a local coordinate system are converted into the host-vehicle's global coordinate system. The sensors' poses are calibrated with respect to the host vehicle's coordinates system. Those reader tasks also perform a pre-processing of raw measurements to produce features (e.g., "L" shape from a point cloud) for object detector or tracker. Our tracking system is designed to run at 100Hz on a single machine on the computing cluster. In practice, however, the operating cycle varies based on the number of features. A typical latency, for example, is around $100\,ms$ in highways and around $200\,ms$ in urban environments. The maximum latency is fixed to $300\,ms$.

For the LIDAR observation model, we used widths and lengths of three objects: OW_{ped}=1m, OL_{ped}=1m, OW_{bike}=1m, OL_{bike}=1.7m, and OW_{veh}=2m, OL_{veh}=5m. For the object management system, we begin to track an object when three consecutive measurements of that object are verified and stop to track the object when no observations are available for $400\,ms$. For the movement classification, we used v_{ped_min}=0.5m/s, v_{bike_min}=1.0m/s, and v_{veh_min}=2.0m/s for *moving* clas-

TABLE II

QUANTITATIVE EVALUATION OF OUR MULTI-SENSOR TRACKING SYSTEM

Dataset Section	Vsion Fusion	Total Seconds	Total Objects	Correctly Tracked	Falsely Tracked	True Positive Rate (%)	False Positive Per Minute
Session 1	w/	900 sec	1,762	1,585	183	89.9	12.2
(w/o RNDF)	w/o			1,466	208	83.2	13.9
Session 2	w/	600 sec	1,371	1,285	57	93.7	5.7
(w/ RNDF)	w/o			1,238	79	90.3	7.9

sification and $d_{ped\_traveled}$=1m, $d_{bike\_traveled}$=2m, and $d_{veh\_traveled}$=4m for *observed moving* classification.

A vision sensor is installed in a forward-looking manner and acquires image frames of 640×480 at 8Hz. Those acquired images are fed to the system over a Gigabit Ethernet interface. To detect three objects (i.e., pedestrians, bicyclists, and vehicles) from images, we used the real-time implementation [4] of the deformable part-based models [8] and produce corresponding vision targets.

To accurately determine the dimensions of objects' models, we computed those objects' pixel height with respect to the distance to our vehicle. From this analysis, we found that the dimension, 72×32, is appropriate to detect pedestrians/bicyclists, reliably up to 13m. For the range between 13m and 26m, we trained a 36×16 pixel-sized model with a HOG cell size of 4×4. Similarly, we trained two rear-view vehicle models, one a 48×48 sized model for the range up to 22m and the other a 16×16 sized model with a HOG cell size of 4×4 for the range between 22m and 55m. Figure 6 shows actual objects' models based on the distance to the vehicle.

B. Experimental Results

Overall, our tracking system showed a good performance on the entire data. For example, when a vehicle is more than 150m away from our vehicle, the tracker begins to track the vehicle with a point model, and is able to switch the point model to a 3D box model when the tracked vehicle is less than 40m from the host vehicle. This is a desirable feature for other modules (e.g., a motion planner) of self-driving vehicles because a host vehicle should know the exact (or approximately close) dimension of a moving object as the objects gets closer to the host vehicle. Figure 7 shows some examples of object tracking. a) and b) show pedestrian and bicyclist tracking results. From these examples, we found that our movement classification worked well to effectively track slow-moving and stop-and-go objects. For the case of c), LIDAR targets were reflected by walls of a tunnel. Because of this, our tracker traced "ghost" targets with a point model. Despite of this, because a vision target was available and associated with the target, our tracker was able to track the target with a 3D box, instead of tracking them with the point model. The cases between d) and h) show some example of vehicle tracking results on city roads and highways.

For the quantitative performance evaluation, it is required to manually label each of the frames in the entire data set. Because this is labor-intensive and error-prone, we evaluate the performance differently. In particular, we had human annotators, using our visualization tool, go over the data second-by-second. While doing so, they counted the number of correctly (and incorrectly) tracked objects. Objects being considered for the evaluation include vehicles in a 150m radius of the host vehicle and pedestrians/bicyclists up to 20m on our vehicle's path. We investigated if the tracking performance is improved when a topological map[1] is given. We also studied how much the performance was improved when the vision target is incorporated. Table II summarizes the experimental results. In short, the detection rate was 93.7% with 5.7 false positives per minute. All result videos for the entire route are available on our project website[2].

VII. CONCLUSIONS AND FUTURE WORK

This paper presented our new moving object detection and tracking system. To improve our earlier system, we redesigned sensor-configuration and installed multiple of radar and LIDAR pairs and three vision sensors. To seamlessly incorporate measurements in different modalities, we revised the previous motion and measurement models and introduced new models for vision measurements. In particular, by using vision's object class and shape information, our tracking system effectively switched between two motion models (i.e., a point and a 3D box models) based on objects' distances to our vehicle. The newly introduced vision targets were also useful to improve the performance of data association and movement classification for measurements from active sensors. Through the test using the data log of actual driving, we demonstrated the improvement and performance gain of our new tracking system.

As future work, we would like to investigate contextual information about urban traffic environments, such asvpresense of lane-markings and side-walks, for improving our tracking system's capability.

VIII. ACKNOWLEDGMENTS

The authors would like to thank the members of GM-CMU Autonomous Driving Collaborative Research Lab for their efforts.

[1] We use a partial Route Network Definition File (RNDF) that only contains the route to the Airport.

[2] http://users.ece.cmu.edu/∼hyunggic/multiSensorFusion.html

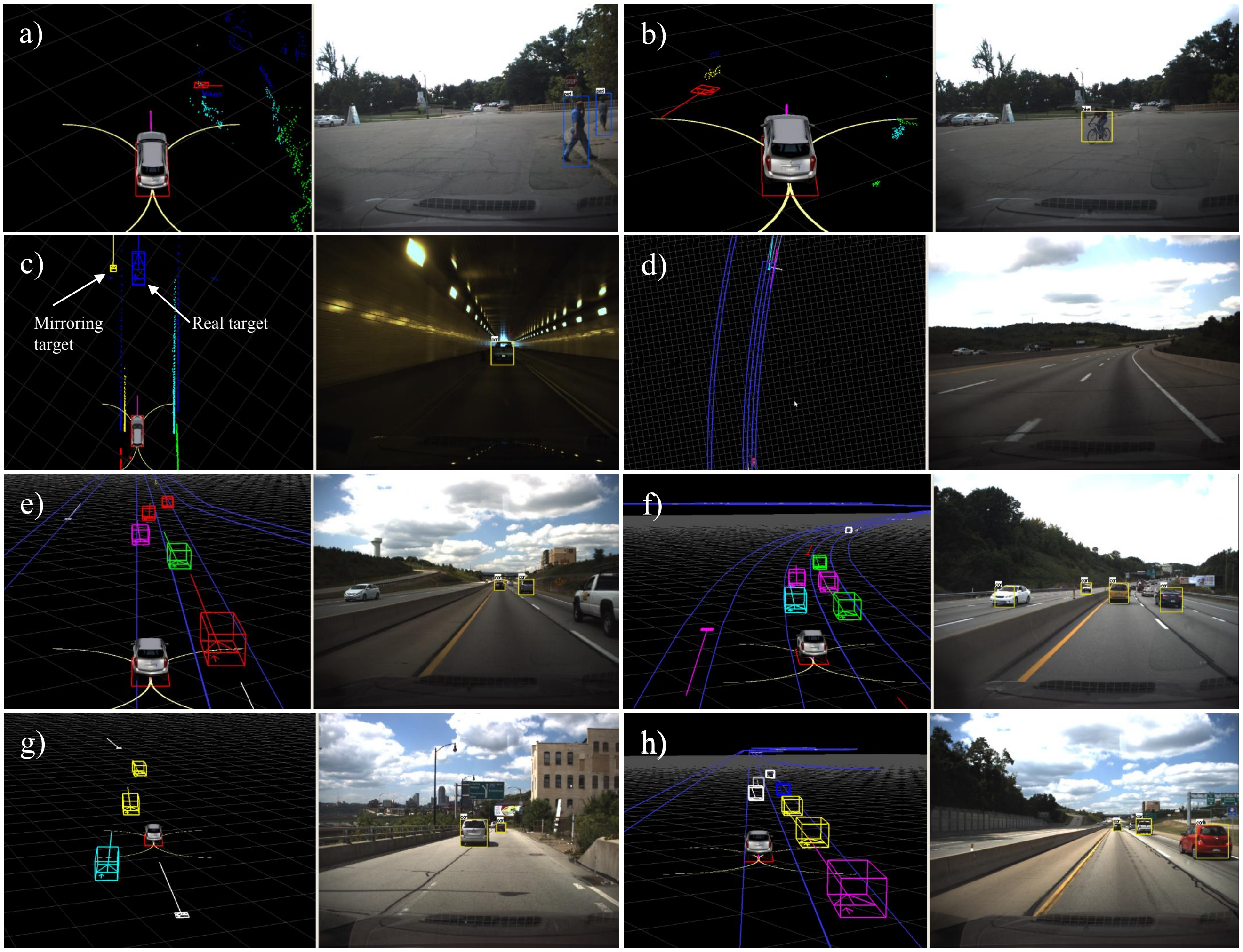

Fig. 7. Some tracking results for the qualitative evaluation. Tracking of a pedestrian (a) and a bicyclist (b), which was enabled by the vision recognition system. (c) Mirroring target issue (see text for the detail). (d) Tracking of a vehicle in far distance. (e)∼(h) Vehicle tracking results in various situations.

References

[1] R. Aufrere, J. Gowdy, C. Mertz, C. Thorpe, C.-C. Wang, and T. Yata. Perception for collision avoidance and autonomous driving. *Mechatronics*, 13(10):1149–1161, December 2003.

[2] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan. *Estimation with Applications to Tracking and Navigation*. Wiley Interscience, 2001.

[3] A. Broggi. The vislab intercontinental autonomous challenge. http://viac.vislab.it/.

[4] H. Cho, P. Rybski, A. B. Hillel, and W. Zhang. Real-time pedestrian detection with deformable part models. In *IV*, 2012.

[5] H. Cho, P. Rybski, and W. Zhang. Vision-based 3d bicycle tracking using deformable part model and interacting multiple model filter. In *ICRA*, 2011.

[6] M. S. Darms, P. Rybski, C. Baker, and C. Urmson. Obstacle detection and tracking for the urban challenge. *IEEE Transaction on Intelligent Transportation Systems*, 10(3), 2009.

[7] H. Durrant-Whyte. *Multi Sensor Data Fusion*. Lecture Notes, 2006.

[8] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 99, 2010.

[9] N. Kaempchen, K. Weiss, M. Schaefer, and K. Dietmayer. Imm object tracking for high dynamic driving maneuvers. In *IV*, 2004.

[10] J. Leonard et al. A perception-driven autonomous urban vehicle. *Journal of Field Robotics, Special Issue on the 2007 DARPA Urban Challenge, Part I*, 2008.

[11] M. Mählisch, R. Schweiger, W. Ritter, and K. Dietmayer. Sensorfusion using spatio-temporal aligned video and lidar for improved vehicle detection. In *IV*, 2006.

[12] C. Mertz et al. Moving object detection with laser scanners. *Journal of Field Robotics*, 30(1):17–43, 2013.

[13] I. Miller et al. Team cornell's skynet: Robust perception and planning in an urban environment. *Journal of Field Robotics, Special Issue on the 2007 DARPA Urban Challenge, Part I*, 25(8):493–527, 2008.

[14] G. Monteiro, C. Premebida, P. Peixoto, and U. Nunes. Tracking and classification of dynamic obstacles using laser range finder and vision. In *IROS Workshop on safe navigation in opern environments*, 2006.

[15] M. Montemerlo et al. Junior: The stanford entry in the urban challenge. *Journal of Field Robotics, Special Issue on the 2007 DARPA Urban Challenge, Part I*, 25(9):569–597, 2008.

[16] C. Premebida, O. Ludwig, and U. Nunes. Lidar and vision-based pedestrian detection system. *Journal of Field Robotics*, 26(9):696–711, 2009.

[17] L. Spinello and R. Siegwart. Human detection using multimodal and multidimentional features. In *ICRA*, 2008.

[18] C. Stiller, J. Hipp, and A. E. C. Róssig. Lidar and vision-based pedestrian detection system. *Journal of Image and Vision Computing*, pages 389–396, 2000.

[19] C. Urmson. The self-driving car logs more miles on new wheels. http://googleblog.blogspot.com/2012/08/the-self-driving-car-logs-more-miles-on.html.

[20] J. Wei et al. Towards a viable autonomous driving research platform. In *IV*, 2013.