Detection and Tracking of Moving Objects at Intersections Using a Network of Laser Scanners

Huijing Zhao, *Member, IEEE*, Jie Sha, Yipu Zhao, Junqiang Xi, Jinshi Cui, *Member, IEEE*, Hongbin Zha, *Member, IEEE*, and Ryosuke Shibasaki, *Member, IEEE*

***Abstract*—In our previous work, we reported a system that monitors an intersection using a network of horizontal laser scanners. This paper focuses on an algorithm for moving-object detection and tracking, given a sequence of distributed laser scan data of an intersection. The goal is to detect each moving object that enters the intersection; estimate state parameters such as size; and track its location, speed, and direction while it passes through the intersection. This work is unique, to the best of the authors' knowledge, in that the data is novel, which provides new possibilities but with great challenges; the algorithm is the first proposal that uses such data in detecting and tracking all moving objects that pass through a large crowded intersection with focus on achieving robustness to partial observations, some of which result from occlusions, and on performing correct data associations in crowded situations. Promising results are demonstrated using experimental data from real intersections, whereby, for 1063 objects moving through an intersection over 20 min, 988 are perfectly tracked from entrance to exit with an excellent tracking ratio of 92.9%. System advantages, limitations, and future work are discussed.**

***Index Terms*—Detection, intersection, laser scanner, moving object, network sensing, tracking.**

I. Introduction

MONITORING traffic behavior at an intersection or collecting traffic data such as speed, motion trajectory, and counts for different types of traffic objects (e.g., car, bicycle, and pedestrian) is of great importance in improving intersection safety and accessibility. Moving-object detection and tracking is a key challenge for this type of application, which has been extensively studied by researchers in the computer vision and the intelligent transportation system (ITS) field. Normally, video cameras are used, as they are less costly, easier to install, and commercially available. An extensive review of the current state of the art in the development of visual surveillance systems can be found in [1], whereas pioneer work can be traced to [2] and [3]. Cameras that are installed on roadsides are used to detect and track pedestrians [4], vehicles [5], or, in general, objects [6] at intersections [7]–[9] or highway scenes [5] to enable estimations of traffic speed [10] and crossing time [11] and to detect traffic events [7], [9]. There are also research efforts that analyze visual data from an airborne camera [12], [13], an on-vehicle camera [14]–[16], or a combination of multiplatform cameras [17]. However, vision-based systems mainly suffer from two problems: 1) occlusion and 2) sudden changes in illumination. In most existing systems, a video camera is required to be set in a particular position, e.g., in a high position, so that the objects on the road can be monitored with less occlusion. However, such a condition may not be easily achieved at many intersections. If additional construction is required, the setup cost may be too high for a nonpermanent system.

In addition to video cameras, laser scanners (also called LiDAR or laser range scanners) are attracting increasing attention in the fields of robotics and ITS, as many intelligent vehicle and moving robotic platforms have laser scanners to assist with driving safety. In such systems, laser scanners are normally set for horizontal scanning to detect and track static and/or mobile obstacles [18]–[20] at a certain horizontal plane with the goal of predicting collisions. As the sensor platform moves, many researchers couple the compensation of the sensor's ego motion (localization), obstacle detection (mapping), and moving-object tracking in simultaneous frames, i.e., Simultaneous Localization And Mapping with Moving Object Tracking (SLAMMOT) [21], [22], for which the pioneering work can be traced to [23]. To constantly monitor an intersection and collect its traffic data, the ITS group at the University of Minnesota, Minneapolis, developed a test bed system that placed a network of radar and laser sensors near a rural intersection [24].

In our previous work [25], we proposed a system and data-processing algorithms for tracking pedestrians using a network of horizontal laser scanners. The system has been successfully used to extract passengers' motion trajectories at subway stations in Tokyo [26]. We recently reported a system to monitor an intersection using a network of horizontal laser scanners [27]. This paper focuses on an algorithm for moving-object detection and tracking, given a sequence of distributed laser scan data of an intersection scene, in which the goal is to detect each moving object that enters the intersection; estimate its state parameters such as size; and track its location, speed, and direction while it passes through the intersection. To the best of the authors' knowledge, this work is unique in that the data are novel, which provides new possibilities but

Manuscript received March 7, 2010; revised May 20, 2011; accepted October 29, 2011. Date of publication December 1, 2011; date of current version May 30, 2012. This work was supported in part by the Hi-Tech Research and Development Program of China under Grant 2007AA11Z225 and in part by the National Natural Science Foundation of China under Grant 90920304 and Grant 60975061. The Associate Editor for this paper was R. W. Goudy.

H. Zhao, J. Sha, Y. Zhao, J. Cui, and H. Zha are with the Key Laboratory of Machine Perception (Ministry of Education), and the School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China (e-mail: zhaohj@cis.pku.edu.cn).

J. Xi is with the State Key Laboratory of Vehicle Transmission, Beijing Institute of Technology, Beijing 100081, China (e-mail: xjq@263.net).

R. Shibasaki is with the Center for Spatial Information Science, University of Tokyo, Chiba 277-8568, Japan.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TITS.2011.2175218

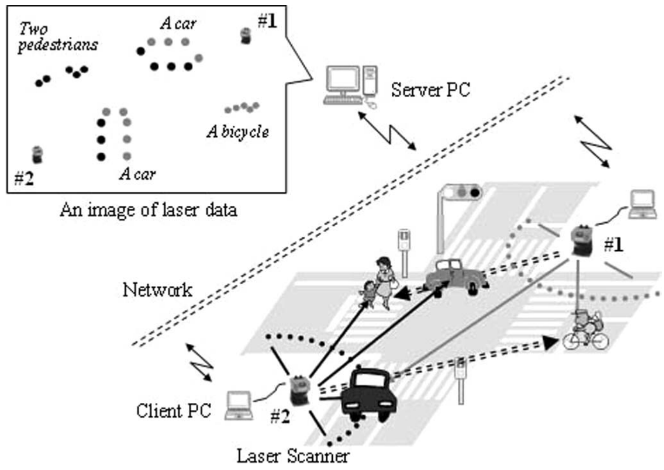


Fig. 1. The system.

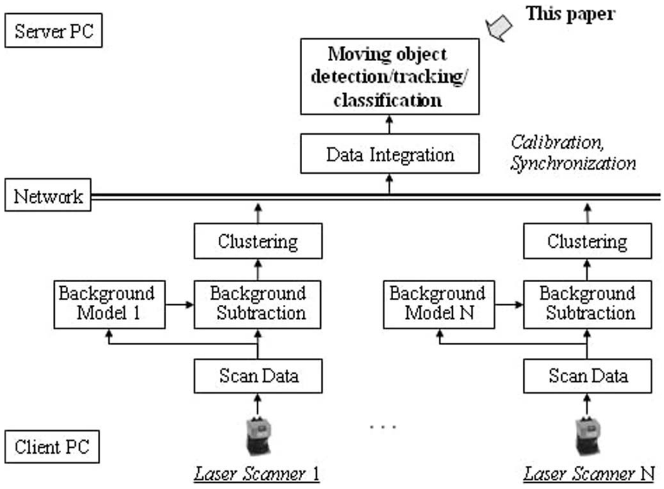


Fig. 2. Flow of major processing modules.

great challenges; this algorithm is the first proposal for using such data in detecting and tracking all moving objects that pass through a large crowded intersection. The experiments demonstrate promising results for such a system to obtain accurate traffic parameters with a low computational cost. All experimental data and results presented here are freely available at our website http://poss.pku.edu.cn.

This paper is organized as follows: In Section II, we give a brief introduction to the system and the data, along with a discussion of their advantages and challenges. In Section III, we propose an algorithm for moving-object detection, followed by an algorithm for moving-object tracking in Section IV. Here, an object model is defined by utilizing the special characteristics of laser scanning; the object model performs a central role in achieving robustness to partial observations and enables data association in crowded situations. Experimental results and discussions are given in Section V, followed by conclusions in Section VI.

II. System and Data

A. Brief Outline to the System and Data

An illustration of the system setting and a flowchart of the major processing modules are given in Figs. 1 and 2, respectively. Laser scanners are set on roadsides, profiling the intersection from different locations. Each scanner is controlled by a client computer, which collects raw measurements from the sensor and performs preliminary processing such as background subtraction and clustering (i.e., extracting the clusters of moving objects) on local scan data. All client computers are connected through a network to a server computer. The server computer periodically broadcasts its time to synchronize the time of all client computers. After setting the sensors, calibration is also conducted by manually matching the data of different sensors to the same static objects. The external geometric parameters within a horizontal plane, i.e., one rotation and two translation parameters from each sensor's coordinate system to a reference frame, with which the laser points from all different laser scanners can be transformed into a common coordinate system, are found. The server computer collects laser scans, as well as local processing results from all client computers, and conducts data integration. An *integrated frame* is generated at each processing interval by collecting the scan of the nearest time stamp from each laser client and converting the coordinates of the laser points to the reference frame. Thus, an integrated frame is an instantaneous cross section of the whole intersection, the laser points of which depict the visible contours of both static and mobile objects in the scanning plane. This type of data integration, which uses a network of laser scanners, has the advantage of covering large intersections and, at the same time, reducing occlusions that occur, particularly in crowded situations. For example, in Fig. 1, a bicycle is blocked by a black car from the viewpoint of laser scanner #2 (dotted line); however, the bicycle is measured by laser scanner #1 (solid line). On the other hand, a pedestrian is blocked by a gray car from the viewpoint of one laser scanner (#1) but is measured by another laser scanner (#2). After data integration at the server computer, an integrated frame that assembles data from all of the laser scanners provides a view that describes a more complete horizontal contour of the moving objects at that moment.

In this work, we use single-row laser scanners called SICK LMS2**. We set the scanning angle to 180° and the scanning resolution to 0.5°; the sensor has a scan rate of approximately 37.5 Hz. Each scan measures 361 range values at an equal angular interval of 0.5°. Range distances might be up to 45 m in a normal traffic scene, with an average range error in the range of 3–10 cm. According to the beam angle, each range value can easily be converted to a 2-D coordinate (called a "laser point") with respect to the sensor's local coordinate system. Laser scanners are set on roadsides and profile the intersection at a horizontal plane of about 40 cm above the ground. Integrated frames are processed at a rate of 10 Hz, considering the traffic speed at an intersection. *An advantage of such a system* is that sensor setting is easy to achieve, i.e., they do not require additional construction or rely on other infrastructure.

B. Advantages of the Data

A sample of laser scan data is shown in Fig. 3. The gray lines, representing road boundaries and safety zones, are sketched for

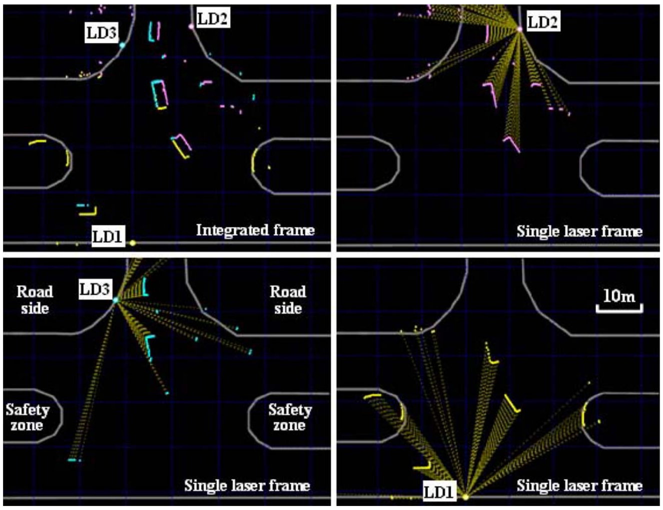


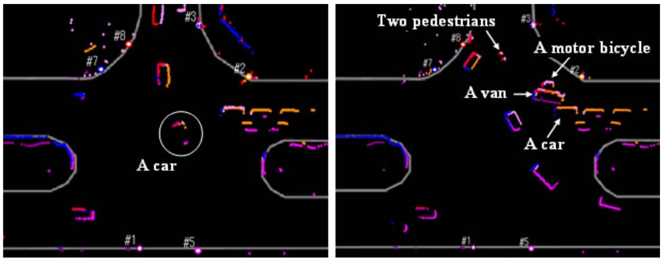
Fig. 3. Data sample demonstrating advantages of the integrated frame.

better understanding of the environment. Three laser scanners are placed on the roadsides of a three-way intersection, whose dimensions are indicated by blue mesh lines in a 10-m step.

An integrated frame that assembles the data from all three sensors is shown on the top left. Laser points are shown in different colors (e.g., pink, yellow, and blue), denoting the data from different laser scanners. As a comparison, the data from three individual laser scanners are also given. Dotted lines are drawn between the sensor and the laser points, demonstrating the laser beams of valid returns. It is obvious that, by integrating the laser measurements from different viewpoints on the roadside, a more complete contour can be recovered for an object that enters the intersection. This idea means that a more accurate estimation of the object state could be achieved by using such a method of integration. Meanwhile, the objects that were blocked or poorly measured by one sensor could be compensated by other sensors. Such cooperation makes it possible to adequately cover a large intersection. The *advantages of the data* are summarized here.

1) The total data size is small (e.g., ∼27 kB/s from one laser scanner), and the data geometry is simple (i.e., a planar Cartesian coordinate system), which makes it possible to perform online fusion of a network of laser sensing data at the raw-data level.
2) 2) A large horizontal coverage can be achieved through the network of sensors.
3) 3) Although a single laser scanner placed at ground level suffers from more occlusion, compared with one at a higher location, it is possible to reduce occlusion through multisensor collaboration, as data from different sensors are complementary.

C. Challenges of the Data



(a) (b)

Fig. 4. Data samples demonstrating challenges of the integrated frame.

Fig. 4 gives two data samples that demonstrate some of the challenges of analyzing the data. Laser points in the red circle of Fig. 4(a) are an assembly of the measurements from different laser scanners to a car. However, these measurements are spatially disconnected. Laser beams on some part of the car failed in the range measurement due to low reflectance material, shallow incidence angles, or other factors. A clustering method might take these measurements to be data from two different objects. On the other hand, Fig. 4(b) demonstrates data from a crowded scene, where two pedestrians walked side by side and a van tried to make a right turn through the narrow path that was generated by a motorcycle and a car, which were waiting for traffic signal.[1] A clustering method might fail to discriminate these vehicles into individual objects, and hijacks[2] could happen in tracking such cluttered data. *Challenges to the use of these data* are given here.

1) The data of a single object might be spatially dispersed, whereas the data of different objects could be spatially close to each other.
2) The data clusters of a single object might have a different appearance, and a single data cluster normally provides only partial information about the object, which may make state estimations inaccurate and data associations ambiguous.
3) Measurement quality are unequal: if an object passing through the intersection is in a central position (where there is a clear view to the laser scanners), it would be completely measured and with high resolution, but if it is in an area where the view to the sensors is blocked, it might be measured only partially and with low resolution.

Integrating the data measurements from different sensors certainly provides more possibilities in detecting the objects in an environment and estimating their states. However, data integration also brings challenges to data handling. To make full use of the data, it is necessary to find an efficient way to associate the data measurements with a single object within a certain integrated frame (“grouping” or “spatial data association”) and along a data stream (“temporal data association”). In addition, a method is required to model the detailed state parameters and achieve an accurate estimation in case of partial observations and crowded situations.

III. Moving-Object Detection

This module finds moving-object candidates (*detections*) in a single integrated frame and extracts feature parameters for an object model.

[1] At many intersections in Beijing, cars turning right do not need to wait for a traffic signal.

[2] Hijack occurs when the label for one object suddenly becomes the label for a different object, as though the second object has stolen, or hijacked, the identity of the first object.

As has been discussed in a previous section, objects might be partially obscured due to occlusions or range failures. For example, when a car is only partially observed on one side, if we do not assume any strong model on the car or we do not even know that the object is a car, an estimation of its size and center point could be quite unreliable. Partial observation could greatly affect the reliability of feature parameter estimation and could subsequently reduce accuracy in tracking moving objects. Developing an object model to support robust feature parameter extraction on partial observation data is key to this procedure, as well as for building a tracking module.

In addition, a desirable detection result is that all moving objects be successfully detected with a minimal number of false alarms. When a moving object enters the area of laser coverage, software can easily find the object as long as it gives a reflection to the laser beam. However, because an object might be simultaneously measured by different laser scanners and because the contour points of an object might be spatially disconnected due to occlusions or range failures, multiple alarms could arise from a single moving object. To reduce multiple alarms and have more complete knowledge for feature parameter estimation, developing an algorithm of grouping the measurements from different laser scanners into the same object is another key to this procedure.

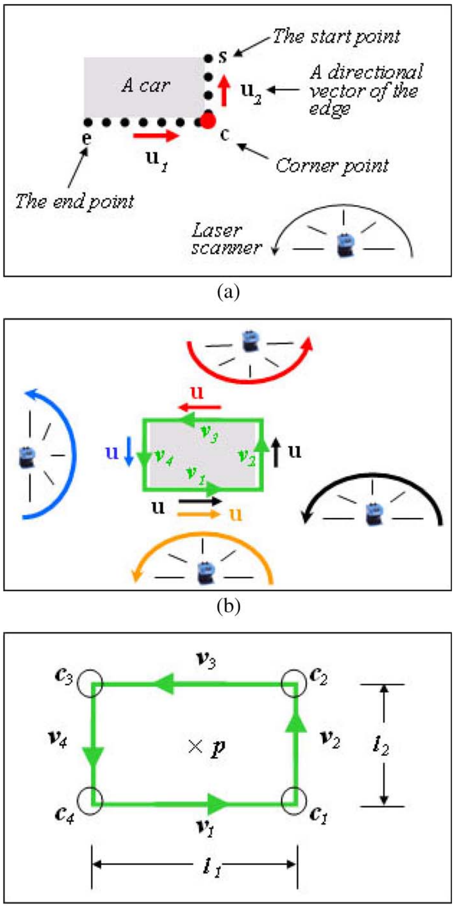
In the succeeding sections, we define an object model, we address a method of grouping the measurements from different laser scanners to detect moving objects in the environment and extract their feature parameters, and we conclude with an experimental result that verifies the algorithm.

A. Object Model

Fig. 5 describes a special characteristic of laser measurements. Suppose that a laser scanner performs counterclockwise scanning, and the horizontal contour of a car is measured by a sequence of laser points from s to e [see Fig. 5(a)]. Simplifying the shape of a car using a rectangle, edges that represent two vertical sides of the car can be detected through a corner detector and a line fitting on the laser points. A directional vector u_i that is associated with each edge is defined according to the scanning order of laser points, e.g., from a point measured later to one measured earlier.

Let u denote a directional vector that is extracted from the data of a single laser scanner after its alignment to a reference frame. We found that no matter where a laser scanner is placed, directional vectors u are equal if they are observations on the same side of the object [see Fig. 5(b)]. Suppose that v_is, $i = 1, \ldots, 4$ are the *directional vectors* defined on each side of a car, and they compose a counterclockwise loop. By matching u with v_is, we can find which side of the object is measured so that the laser points that correspond to u are used to update the estimates of that specific side. In this work, we call u a *support vector* of the side v_i.

Based on the aforementioned considerations, an object model is defined in this work with the feature parameters shown in Fig. 5(c), where the shape of an object is simplified using a rectangular model. Developing a more accurate model for each kind of object will be addressed in future work. In addition, a reliability item is defined for each feature parameter for the sake of partial observations (see Table I). Currently, reliabilities are estimated with binary values, i.e., true $= 1$ or false $= 0$. In the case of a directional vector, the reliability denotes whether the side has a support vector (*reliable*) or not (*unreliable*). In the case of a corner point, if both neighboring sides are supported, the corner point is a (*reliable*) one; otherwise, it is a guess through other feature parameters on the object model (*unreliable*). In the case of dimensional size, reliability tells whether the corresponding feature parameter represents a full dimensional size (*reliable*) or perhaps a partial one (*unreliable*). In the case of a center point, which cannot be directly observed, the reliability denotes whether the coordinates are estimated from other *reliable* feature parameters. A more detailed description of each parameter in an object model can be found in Appendix A.



(a)

(b)

(c)

Fig. 5. Definition to an object model. (a) Measurement to a car from a single laser scanner. (b) Measurements to a car from a network of laser scanners. (c) Feature parameters of the object model.

TABLE I
PARAMETERS IN AN OBJECT MODEL

Item	Feature	Reliability
directional vectors ($i = 1, .., 4$)	v_i	rv_i
corner points ($i = 1, .., 4$)	c_i	rc_i
a center point	p	rp
dimensions on two vertical sides ($i = 1, 2$)	l_i	rl_i

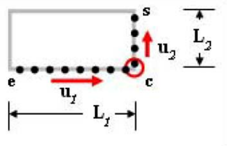
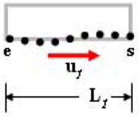
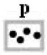
Type	*Sketch*	*Feature Parameters*
Two Axes:	 s $\mathbf{u}_2$ $\mathbf{L}_2$ e $\mathbf{u}_1$ c $\mathbf{L}_1$	**c** : *corner point* $\mathbf{u}_1$: *a directional vector from e to c* $\mathbf{u}_2$: *a directional vector from c to s* $\mathbf{L}_1$: *length between e and c* $\mathbf{L}_2$: *length between c and s*
One Axis:		$\mathbf{u}_1$: *a directional vector from e to s* $\mathbf{L}_1$: *length between e and s*
Zero Axis:		**p** : *center point*

Fig. 6. Feature parameters extracted from a data cluster.

B. Data Grouping and Moving-Object Extraction

The input of this procedure is a set of data clusters in an integrated frame; the data clusters are first extracted through background subtraction and clustering on local scan data at each client computer, then uploaded to a server computer through a network, and finally transformed and assembled into a reference frame so that a data cluster is a measurement to a moving object from a single laser scanner. However, a single moving object might be simultaneously measured by a number of clusters. This procedure is to associate the data clusters of the same moving objects, extract feature parameters, and evaluate the reliabilities for each moving object according to the object model previously defined.

Preprocessing is first conducted to extract feature parameters from each data cluster as follows: Through a KL transform on the laser points in each data cluster, clusters are divided into three types according to the number of obvious axes that can be detected. Feature parameters are then extracted accordingly, as shown in Fig. 6. For a two-axes cluster, a corner detector is first conducted to divide the laser points into two sequential parts, and line fittings are then conducted on each part to extract the directional vectors u_1, u_2 and edge lengths L_1 and L_2. For a one-axis cluster, a directional vector u_1 and a length L_1 are extracted, leaving other parameters invalid. If no obvious axis is detected, i.e., there is a zero-axis cluster, then a center point is estimated.

Moving-object detections are then extracted one by one until all of the clusters have been associated in the flow of Fig. 7, where a detection g_i contains a group of clusters $\{c_i\}$ that have been associated with it, as well as an estimation of object model m_i. In extracting each detection, from the clusters that have not been associated with any detections, a cluster c_i is chosen to initialize a new detection g_i. Normally, an edge is longer, and estimation of the directional vector is more reliable. We chose cluster c_i, which has the longest edge length, to initialize a new detection. A function of estimating an object model on a set of clusters is detailed in Appendix B. Detection g_i then iteratively absorbs other free clusters. In each iteration, g_i finds a cluster c' in its vicinity that has not been associated with any detections. A new object model m_i' is estimated on a merged group of clusters $\{c', c_i | i = 1, \ldots\}$. If successful, c' is associated with g_i, i.e., is added to the group of $\{c_i\}$, and m_i is replaced by m_i'. The iterations continue until no more clusters can be associated with g_i.

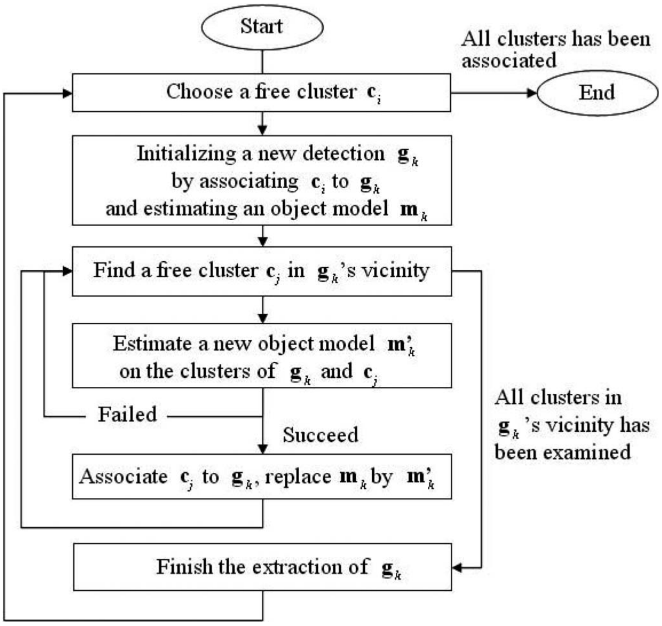
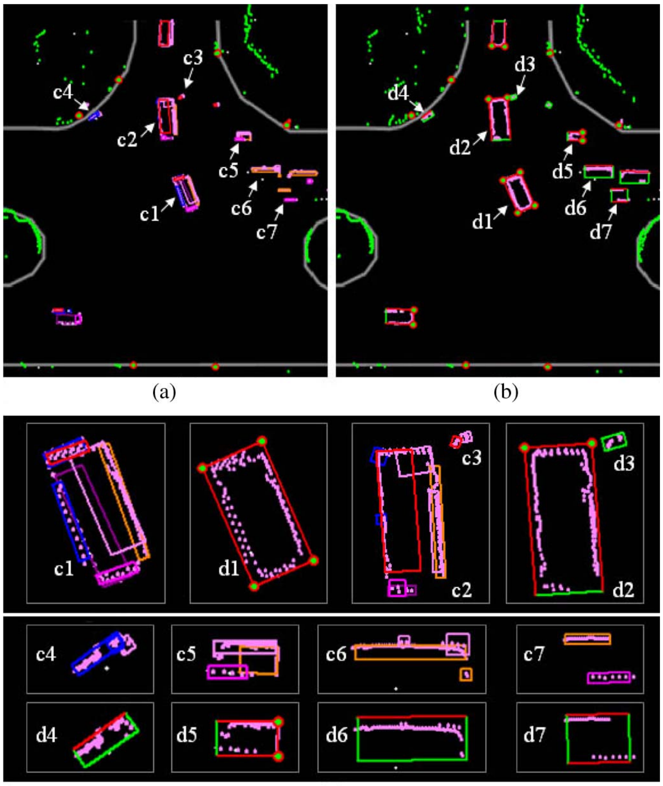

Start
All clusters has been associated
Choose a free cluster $\mathbf{c}_i$
End
Initializing a new detection $\mathbf{g}_k$ by associating $\mathbf{c}_i$ to $\mathbf{g}_k$ and estimating an object model $\mathbf{m}_k$
Find a free cluster $\mathbf{c}_j$ in $\mathbf{g}_k$'s vicinity
Estimate a new object model $\mathbf{m}'_k$ on the clusters of $\mathbf{g}_k$ and $\mathbf{c}_j$
Failed
Succeed
All clusters in $\mathbf{g}_k$'s vicinity has been examined
Associate $\mathbf{c}_j$ to $\mathbf{g}_k$, replace $\mathbf{m}_k$ by $\mathbf{m}'_k$
Finish the extraction of $\mathbf{g}_k$

Fig. 7. Flow of extracting moving-object detections. Input: a set of clusters $\{c_i\}$. Output: a set of moving-object detections $\{g_k\}$.



(a) (b)

(c)

Fig. 8. Experimental result of moving-object detection. (a) Clustering. (b) Detection. (c) Enlarged figures.

C. Detection Result

Fig. 8 shows an experimental result for moving-object detection. The data clusters that are extracted from each individual laser scan are shown in Fig. 8(a), where clusters are represented using rectangles that are recovered with their feature parameters; different colors are randomly assigned to discriminate the clusters. The set of moving-object detections is shown in Fig. 8(b), in which each object detection is represented by a rectangle, the shape of which is recovered with the feature parameters. Edge colors and circular markers are denoted for reliability items. If a side is supported by observation data with its reliability item set to "true," then it is drawn in red and, otherwise, in green. If a corner point is detected with its reliability item set to "true," it is drawn with a circular marker and, otherwise, with no marker. A number of objects are picked up, and their results are enlarged in Fig. 8(c). The selected objects are numbered with a header character of either "c" or "d" for the result of "clustering" or "detection," respectively. Note that the sequential number in Fig. 8 is associated with neither a cluster nor a detection but with a moving object instead. The detections are resulted after the tracking module, where some of them might be merged to solve confusions (see the next section for details).

We will next explain the result for each object. Object #1 is a fully observed car. Many data clusters are extracted for the car (c1); some of these clusters are duplicated measurements of the same sides and largely overlap. A major task of moving-object detection is to associate the data clusters with the same object and to fit the parameters of the object model. As a result, a detection (d1) is obtained, where each side is shown in red, meaning that the side is supported by observational data. Each corner is shown with a red circle, meaning that the corner is reliably detected; the size and orientation of the rectangle are recovered exactly according to the parameters of $l_{1,2}$, $v_{1,\ldots,4}$, and p. Object #2 is a car that is not completely observed. In detection result (d2), one side is shown in green, which represents the lack of a support vector for this side. Two corner points at the intersections of the red sides have circular markers, whereas the other two corner points are blank, meaning that extractions of the latter corner points might not be true. Object #3 is a pedestrian standing on a cross road, waiting for car #2 to pass by. Two small clusters (c3) are individually extracted from each laser scanner; a detection (d3) with no support vector and no reliable corner point is finally obtained. Similarly, object #4 is a bicycle; its detection has one supported side and no reliable corner point. Object #5 is a motorcycle; its detection has three supported sides, with two reliable corner points. Object #6 is a car; its detection has one supported side, with no reliable corner point. Object #7 is a car with two opposite sides supported by observation; no corner point is reliably detected. Although some objects are not rectangular in shape and even though the contour of a car is not exactly a rectangle, rectangles recovered in the parameters of an object model serve as bounding boxes for the data of moving objects.

IV. Moving-Object Tracking

A set of moving-object detections $G^k = \{g_1^k, g_2^k, \ldots, g_n^k\}$ is extracted from each integrated frame k and then forwarded to tracking modules as the observational input at each frame to track the data from moving objects and to estimate their state parameters, including shape, size, location, speed, and direction along a stream (called "track"). Here, we first define the problems in moving-object tracking, and then, we address our solutions; afterward, we give an experimental result that verifies the algorithm.

A. Problem Definition

1) State Estimation: Given a sequence of observations $G_i^k = \{g_i^1, g_i^2, \ldots, g_i^k\}$ for the moving object i from frame 1 to k, the objective is to find a track $T_i^k = \{t_i^1, t_i^2, \ldots, t_i^k\}$ for which the problem of estimating state t_i^k at each frame k can be probabilistically defined as $p(t_i^k|G_i^k)$ and further extended to

$$\begin{aligned} P\left(t_i^k|G_i^k\right) &= p\left(t_i^k|g_i^k, G_i^{k-1}\right) \\ &=_{\text{Markov}} \quad p\left(t_i^k|g_i^k, t_i^{k-1}\right) \\ &\propto_{\text{Bayes}} \quad p\left(g_i^k|t_i^k\right) \cdot p\left(t_i^k|t_i^{k-1}\right). \end{aligned} \tag{1}$$

A solution to t_i^k uses the Maximum *a Posteriori* method as follows:

$$t_i^k = \arg\max_{\hat{t}_i^k} p\left(g_i^k|\hat{t}_i^k\right) \cdot p\left(\hat{t}_i^k|t_i^{k-1}\right) \tag{2}$$

where $p(\hat{t}_i^k|t_i^{k-1})$ is a prediction of a state transition, and $p(g_i^k|\hat{t}_i^k)$ is a likelihood measure, meaning that, if an object is at a state $\hat{t}_i^k$, the probability that an observation g_i^k is obtained.

2) Data Association: In a crowded situation, associating observations with their tracks of moving objects along a data stream is key to correct state estimation. In contrast with that in the detection module, which groups the measurements of different laser scanners to the same objects in one instantaneous measurement (an integrated frame), data association in the tracking module is conducted along a temporal axis.

For any pair of a track i with its state t_i^{k-1} at a previous frame $k-1$ and an observation g_j^k at the current frame k, an association candidate $\langle (i,j), t_{ij}, d_{ij} \rangle$ can be generated, where t_{ij} is a state update by associating observation j to track i, and d_{ij} is a measure for their association probability. The values of t_{ij} and d_{ij} are estimated by extending (2), i.e.,

$$t_{ij} = \arg\max_{\hat{t}_i^k} p\left(g_j^k|\hat{t}_i^k\right) \cdot p\left(\hat{t}_i^k|t_i^{k-1}\right) \tag{3}$$

$$d_{ij} = p\left(g_j^k|t_{ij}\right) \cdot p\left(t_{ij}|t_i^{k-1}\right). \tag{4}$$

Given a set of tracks $\{T_i^{k-1}\}$ with their states updated to the previous frame $k-1$ and a set of observations $\{G_j^k\}$ at the current frame, data association is to find a set Φ^k, i.e.,

$$\sum_{(i,j)\in\Phi^k} d_{ij} \rightarrow \max \tag{5}$$

where, for any track i or observation j, it can only appear at Φ^k once. Subsequently, the state of a track i at frame k is updated using a prediction hypothesis t_{ij} as follows:

$$t_i^k \leftarrow t_{ij} \qquad \text{if } \exists j, \quad (i,j) \in \Phi^k. \tag{6}$$

3) Analytical Solutions: Given a previous state t^{k-1} of a moving object and an observation g^k at the current frame, we

TABLE II
FIVE CASES IN HYPOTHESIS GENERATION: DEFINITION

Case	State t^{k-1}	Observation g^k	
	support vector	support vector	valid corner point
SS1	O	O	O
SS2	O	O	×
SS3	O	×	×
SS4	×	×	×
SS5	×	O	×

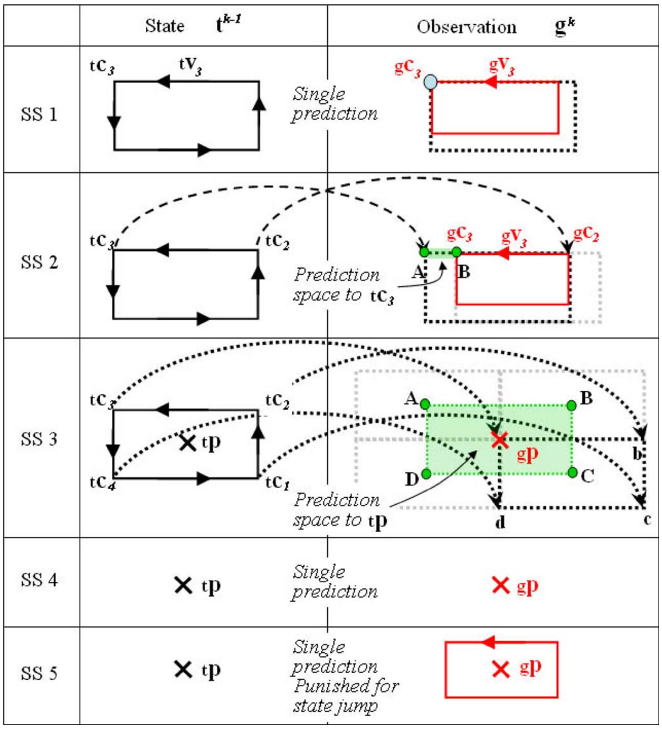


Fig. 9. Five cases in hypothesis generation.

need to find analytical solutions to the following items to estimate t_{ij} and subsequently estimate d_{ij} in the aforementioned equations.

1) Generate a set of prediction hypotheses $\hat{t}^k$.
2) Evaluate the probability of a state transition $p(t^k|t^{k-1})$.
3) Estimate the likelihood measure $p(g^k|t^k)$.

In addition, we also need to find a solution to Φ^k, in which tracks and observations are associated.

Here, we discuss each of the solutions in detail. At this section, we parameterize the state of a track at each frame by using the object model that is defined in the previous section, as well as by using the motion parameters of the interframe speed and direction. For discrimination, a header character "t" or "g" is associated with the parameters of the object model in either the state or the observation.

B. Hypothesis Generation

A state transition can be estimated by binding a pair of corresponding points between t^{k-1} and g^k along with a pair of directional vectors, whereas the number of hypotheses $\hat{t}^k$ are dependent on how reliably these correspondences can be established. In this work, a method is developed by restricting hypothesis generation to state observation matching, and five cases are studied with different combinations of t^{k-1} and g^k. Definitions and illustrations of the five cases are given in Table II and Fig. 9, respectively. Note that a parameter alignment is conducted beforehand so that the parameters in t^{k-1} match with parameters that have the same subscript in g^k. In the first three cases SS1–SS3, state t^{k-1} has at least one support vector, which indicates that establishing correspondence between supported directional vectors is reasonable. However, in the last two cases SS4–SS5, state t^{k-1} has no support vector, telling us that the state estimation up to frame $k-1$ suggests a small-scale object such as a pedestrian, the shape of which should be described using a point rather than a rectangle to establish point correspondences only. Cases are also divided in the situation of g^k, yielding different methods in hypothesis generation. For example, as shown in case SS1 in Fig. 9, g^k has a valid corner point gc_3 and a support vector gv_3. By projecting tc_3 and tv_3 of t^{k-1} onto gc_3 and gv_3, a hypothesis can be generated of a dotted rectangle with high reliability. As a result, for case SS1, a single hypothesis is generated on the matched pairs of valid corner points and support vectors. On the other hand, in case SS2 in Fig. 9, the side of gv_3 is supported by observations, whereas two ends are not valid corner points, indicating that a partial observation might occur. By projecting tv_3 to gv_3, the dotted rectangle can slide along gv_3 within a certain range, suggesting a number of hypotheses. For example, a number of hypotheses for tc_3 can be generated by picking random samples between B and A, where point A is a prediction for tc_3 if aligning the other end of side tc_2 to gc_2. However, in this case, the hypotheses are considered to have low reliability due to occlusion. Hypothesis generation in cases SS3–SS5 is developed similarly; their details are given in Appendix C.

C. Likelihood Measure and State Transition

The reliability of each hypothesis, as previously discussed, reflects the matching between the state and the observation data appearance. The reliability is used as the likelihood measure $p(g^k|\hat{t}^k)$ in this work, and its value is empirically assigned.

As in this work, a method of hypothesis generation is developed by restricting the prediction space through state-observation matching, where a state transition hypothesis is generated by binding a pair of corresponding points along with a pair of directional vectors. Let $\langle p, \hat{p}\rangle\langle v, \hat{v}\rangle$ be the corresponding pairs between t^{k-1} and $\hat{t}^k$; consistency of the hypothesized state transition is evaluated by comparing them with a linear dynamic model on a Gaussian. Let p^* and v^* be the predictions of p and v in a linear dynamic model; then, the state transition probability is estimated as follows:

$$p(\hat{t}^k|t^{k-1}) = G\left(\omega_1 * d_1(\hat{p}, p^*) + \omega_2 * d_2(\hat{v}, v^*)\right) \qquad (7)$$

where d_1 estimates the Euclidean distance between $\hat{p}$ and p^*, and d_2 finds the angle between $\hat{v}$ and v^*. The values of ω_1 and ω_2 are the weights on experience.

D. Data Association

A matrix (called *association matrix*) is used to describe the problem, as shown in Table III, where each item is measure

TABLE III
ASSOCIATION MATRIX

	t_1^{k-1}	$\ldots$	t_i^{k-1}	$\ldots$	t_n^{k-1}
g_1^k					
$\ldots$					
g_i^k			d_{ij}		
$\ldots$					
g_m^k					

d_{ij} [see (4)] if associating observation j with track i. Ideally, each row and each column has only one valid value, meaning that the object can only be associated with the observation, and vice versa. However, we do not always have such perfect track-observation matching in a cluttered environment, as a single column or a single row might have a number of valid values, causing confusion in data associations. Here, we discuss solutions to reach a result, where each row and each column has no value or one valid value. Here, if column i has no valid value, then the observation of the moving object i might be totally occluded or the object might have gone out of the laser scanner's measurement range; if row j has no valid value, then there might be a new object that entered the scene.

CS 1: A column has multiple valid values, i.e., a number of observations can be associated with the track of a single moving object. This confusing situation can be further divided into two cases.

1) The observations are correct; a number of moving objects have been tracked as one during the past frames (*merge* in tracking).
2) The track of a moving object is correct; measurements to the object are split into a number of observations (*split* in detection).

The solution to this case is that we try to merge observations using the algorithm listed in Appendix B. The data clusters of all observations are used to generate an object model. If successful, it is considered to be in CS 1.2: The observations are merged, and the association matrix is regenerated and solved from the beginning. If it fails, it is considered to be in CS 1.1: Associate the observations with the largest probability of the track and generate new moving-object tracks for others.

CS 2: A row has multiple valid values, i.e., a number of moving-object tracks can be associated with a single observation. This confusing situation can also be divided into two cases.

1) The observation is correct; a single moving object is tracked as a number of individuals during the past frames (*split* in tracking).
2) The moving-object tracks are correct; measurements to the objects are merged into a single observation (*merge* in detection).

A solution to this case is that we try to merge the tracks during the past frames. At each frame, we try to merge the observations using the algorithm listed in Appendix B. If successful for all past frames, it is considered to be in CS 2.1: The tracks are merged, and the association matrix is regenerated and solved from the beginning. If it fails, the case is considered to be in CS 2.2: Update the states of moving objects according to a linear dynamic model.

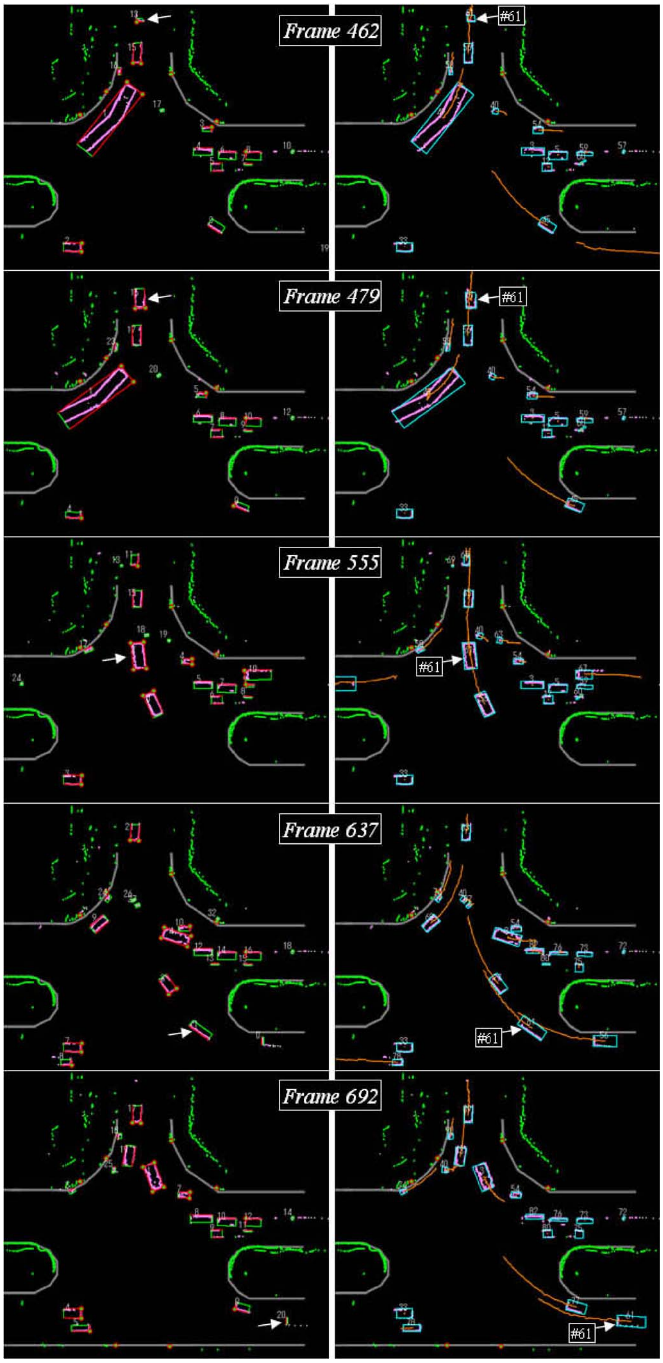


Fig. 10. Tracking result.

E. Tracking Result

Fig. 10 shows the algorithm of moving-object tracking through a sequence of results, where, at an iteration of each frame, the input is a set of detections $G^k = \{g_{1,\ldots,m}\}^k$ (shown in the left column of Fig. 10), as well as a set of tracks $T^{k-1} = \{t_{1,\ldots,n}\}^{k-1}$, with their states updated to the previous frame. Functions of this module are to update the state of existing tracks, create tracks for newly detected moving objects, and delete tracks when they expire to output a new set of tracks $T^k = \{t_{1,\ldots,n'}\}^k$, with their states updated to the current frame (shown in the right column of Fig. 10). Tracking results are demonstrated as follows: Each track is represented using a

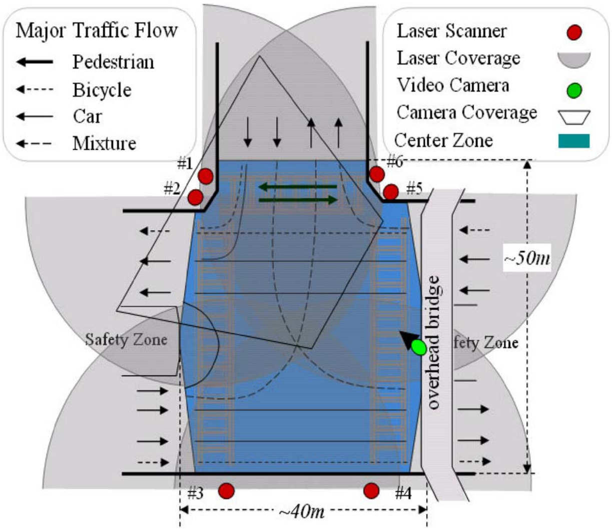


Fig. 11. Sketch of the experimental site and sensor setting.

rectangle in blue, which is recovered from the state parameters at the current frame; the trajectory of its center point over the last 50 frames (5 s) is shown with an orange line, the length of which reflects the moving object's speed.

The sequential results in Fig. 10 present the tracking of car #61 from when it enters the intersection to when it leaves. At the beginning, when car #61 enters the intersection, only its front side is measured, yielding a small rectangle in both detection and tracking results at frame #462. As the car moves forward to the center area of the intersection, more sides are measured with higher resolution. At frame #479, three sides of the car, i.e., the right, left, and front are simultaneously measured, yielding a more reliable detection and a more accurate estimation of the car's state parameters. At frame #555, the car is fully measured, and in its detection result, four sides are supported, and four corner points are reliably detected; in the tracking result, the recovered rectangle reaches its maximum size. As the car leaves the scene, a partial measurement is taken again. At frame #637, the car is detected with one support vector and no reliable corner point; a thin rectangle is recovered in its detection result. However, in the tracking result, the blue rectangle kept its previous size. This event happens because the reliability items (rl_i) in the detection result indicate that the observation parameters are less reliable, and therefore, the corresponding state parameters related to the moving-object track are not updated. At frame #692, only the rear side is measured. However, as the only side supported in the detection result is correctly recognized as the rear side, the car can still be tracked with rather high accuracy.

V. Experimental Results and Discussions

We study the performance of the algorithm for moving-object detection and tracking using experimental data that were collected at a three-way intersection near the campus of Peking University. A sketch of the intersection, as well as sensor layouts, is shown in Fig. 11 and a picture of the experimental site is shown in Fig. 12. We define a center zone of the intersection, as shown in blue in Fig. 11, that is more important for measurement. The center zone has a dimension of approximately $40 * 50\ \mathrm{m}^2$. Some major traffic flows are also drawn from experience to describe the traffic situation; however, these flows are not used as *a priori* knowledge in the detection and tracking of the moving objects. Six LMS2** are set on the sides of the road. The major area covered by each laser scanner is shown as a semicircle with a radius of about 30 m. The value is set from experience, considering both the sensor's capability and the crowdedness of the intersection. In this experiment, each laser scanner is set on a small chair, scanning at a horizontal plane of about 40 cm above the local ground surface. It is better to let the sensors profile in the same horizontal plane; however, in a real situation, this requirement is difficult to meet with high accuracy. We also examine how critically such an alignment error can affect the final results. Each laser scanner is controlled by a notepad (Thinkpad T42) and powered using a car battery. The notepads are synchronized by a server computer before data collection. During the experiments, each laser scan is recorded along with a time log at the client computer's clock. Integrated frames are then generated at equal intervals (100 ms in this work) by assembling the laser scans of the same time log (within a range of half a scan cycle ± 13 ms in this work) from different sensors. As the objective of this experiment is to study the algorithmic performance of moving-object detection and tracking, raw laser scans are recorded on site, and additional data processing is conducted in offline modes. To perform validation, a video camera is set on an overhead bridge near the intersection. Because of the limited viewing angle and elevation of the camera, only a part of the intersection is covered by the camera, as shown in Fig. 11.

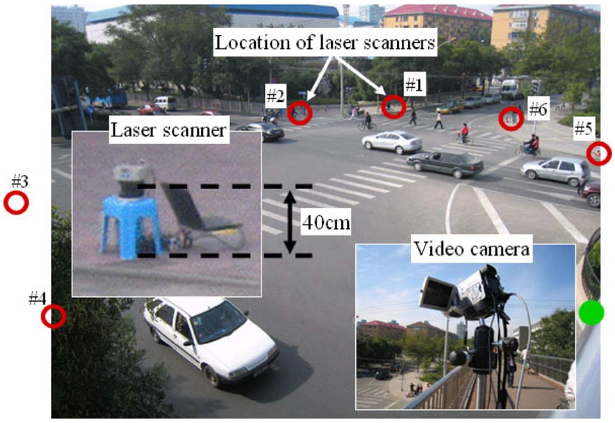


Fig. 12. Experimental site.

Here, we study results from different traffic situations, examine the results by backprojection on video, present some quantitative evaluations, and provide a discussion of the limitations of this work and the directions for future work.

A. Case Studies

Some successful detection and tracking results have already been previously presented (see Figs. 8 and 10). As their purpose was mainly to verify the algorithms, the results were for relatively simple situations. In this section, we study results in more complicated situations. We also intend to find whether the results have the accuracy to reflect interactions among different objects in crowded situations; therefore, Fig. 13 is summarized accordingly. From frame #584 to #614, two pedestrians (#40 and #65) on a zebra zone are detected and tracked. At frame #584, pedestrian #40 waits for car #63 to pass by; however, another car #67 is approaching. Pedestrian #40 waits in place for car #67 to pass through at frame #592; meanwhile, pedestrian #65 approaches pedestrian #40 from the rear. After car #67 passes by at frame #612, pedestrian #40 starts to move, and pedestrian #65 catches up to pedestrian #40 at frame #614. However, there are still other cars passing in front of them, e.g., #70. From frame #614 to #667, the result of an extremely crowded situation is presented. A motorcycle #56 and a car #83 are waiting for a traffic signal. A van #3 turning right tries to pass between them. Recall that, as stated earlier, in intersections in Beijing, right-turn vehicles do not need to wait for traffic signals. It can be seen at frame #614 that the three objects are quite close; data from the three objects are almost merged together. However, the detection method correctly discriminates their individual observations and accurately fits their parameters on the object models (see the left column for the detection results). This result successfully demonstrates that the object model and the detection algorithm presented in this work efficiently perform, even in crowded situations. As the van moves along the narrow path between the motor bicycle and the car and finally goes through the intersection at frame #667, its state is correctly tracked, as are the states of the other two vehicles.

We need to stress that this algorithm does not require that all sides of an object be simultaneously measured. As an object moves along, different sides are measured. The object model can tell which sides of the object are measured by matching laser points on the object model. Such knowledge can be integrated over time while tracking a moving object; the state parameters that correspond to each side are updated only when measured. In this manner, a temporary occlusion or partial observation does not affect the final estimation of an object's true state.

B. Video-Based Examination and Comparison

Video streams are simultaneously recorded with laser measurements for result validation. A synchronization and calibration are conducted before data acquisition to find parameters that are shared between video and laser integrated frames, so that the laser-based moving-object detection and tracking results at each integrated frame can be backprojected to the video image with the same time stamp. (Here, we do not describe in detail the synchronization and calibration procedures, as they are beyond the scope of this paper.) A sequence of results is shown in Fig. 14. For comparison, (right column) the laser processing results, as well as (left column) their backprojections onto video images, are both demonstrated. From the video images, it can be seen that the scene is quite complex: there are many different types of objects, different motion patterns, and large occlusions. At frame #21500, a bus (#443), a bicycle (#437), and a truck (#453) are crossing the intersection. At frame #21590, the bus completely blocks the camera's view of

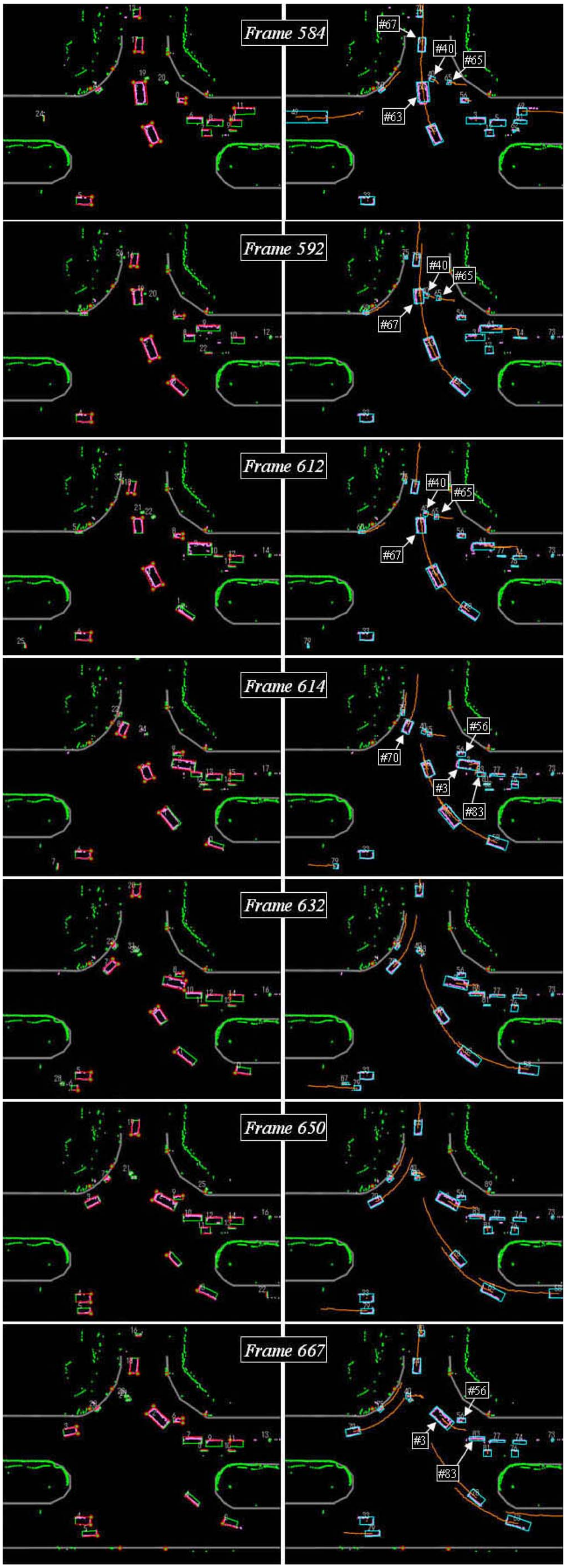


Fig. 13. Successful detection and tracking results.

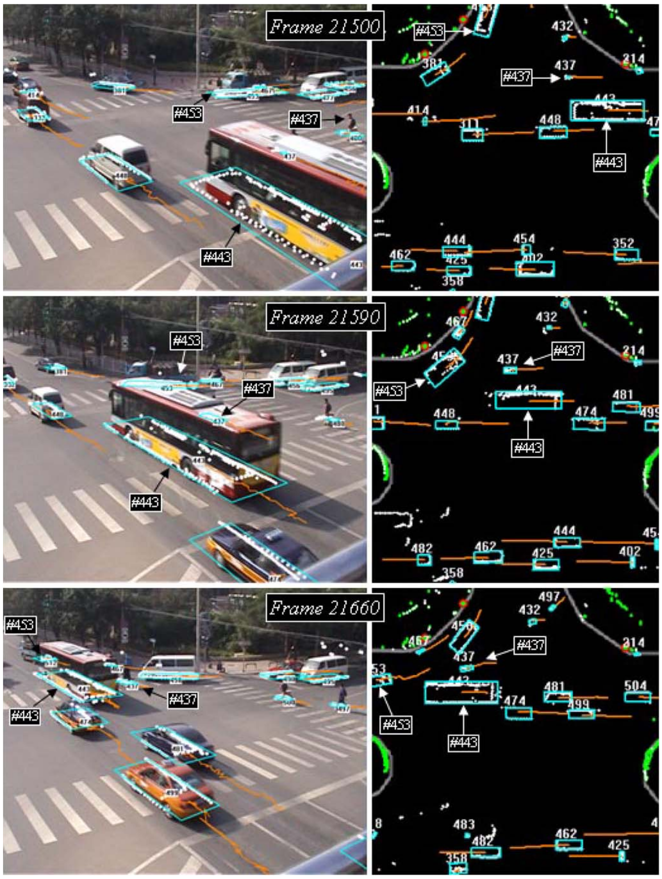


Fig. 14. Examination by overlapping the laser-based processing results on video.

the bicycle, and the truck is also heavily occluded. At frame #21660, the bus overtakes the bicycle from the camera's view, whereas only a small part of the truck can be seen in the image. The heavy occlusions and data overlap bring great challenges to video-based processing. However, it can be seen that, with laser data, the objects are more easily discriminated. Although occlusions happen, as long as the object is not blocked from the view of all laser scanners, the object can be detected and tracked. In Fig. 14, the bus (#443), bicycle (#437), and the truck (#453) are all successfully detected and tracked using the proposed laser-based algorithm.

C. Quantitative Examination

Quantitative examinations are conducted of the detection and tracking results at the center zone of the intersection, which is shown in Fig. 11.

1) Detection Results: Detection results are examined over 1000 frames (100 s of data). At each frame, the detection result of each moving object votes for either of the following four counters: 1) perfect; 2) split; 3) merged; and 4) none, where "perfect" counts successful detections; "split" is voted if a single object has a number of detections; "merged" is the reverse (i.e., a detection corresponds to a number of objects); "none" means that, according to the previous or successive frames, there should be an object. However, no detection is obtained due to either lack of measurement data or failure in detection. In addition, moving objects are divided into pedestrian, car, and bicycle, where bicycle includes two-wheeled bicycle, two-wheeled motorcycle, three-wheeled cycle, and more categories. Results for each type of object are individually counted by an operator, as shown in Table IV, where "total" is the sum over four counters. Considering that, in either case of "correct," "split," or "merge," an object is detected but it might be imperfect, the detection ratio "d.ratio" is calculated as "(correct+split+merge)/total." On the other hand, considering that both "split," "merge," and "none" are erroneous or failed detections, the perfect ratio "p. ratio" is calculated as "perfect/total." It can be found from the table that the results for cars and bicycles are excellent, as the perfect ratios are above 0.9. As for pedestrians, the detection ratio is counted up to 0.91; however, as there are many merge and none cases, the perfect ratio is only 0.551. If a pedestrian is near to other objects, the pedestrians' data might be merged with them, yielding a single detection. This situation can be found at frame #671 of Fig. 16, where the data of two pedestrians closely walking, i.e., #40 and #63, are merged together for a long period, yielding that the track for pedestrian #63 expires. Here, we examine how the imperfect and failure detections affect the final tracking results.

TABLE IV
QUANTITATIVE EXAMINATION OF DETECTION RESULTS DURING 1000 FRAMES

type	perfect	split	merge	none	total	d.ratio	p.ratio
car	6915	614	7	89	7625	0.988	0.907
bicycle	1571	82	0	24	1677	0.986	0.938
pedes.	799	13	508	130	1450	0.910	0.551
sum.	9285	709	515	243	10752	0.977	0.864

TABLE V
QUANTITATIVE EXAMINATION OF TRACKING RESULTS DURING 20 MIN

type	perfect	broken	error	total	t.ratio	p.ratio
car	636	22	17	675	0.975	0.942
bicycle	322	8	21	351	0.940	0.917
pedes.	30	2	5	37	0.865	0.811
sum.	988	32	43	1063	0.960	0.929

2) Tracking Results: Tracking results are examined over 20 min (12 000 frames). For each moving object, the tracking results from entering to leaving the center zone of the intersection votes for either of the following three counters: 1) perfect; 2) broken; or 3) error. The identification "perfect" counts successful tracking; "broken" is voted if tracking to a moving object expired halfway, because, even when it is tracked again, a different ID is assigned to the new track; and "error" is counted if a merge or hijack happened, yielding an erroneous track. Moving objects are also divided into three groups, i.e., pedestrian, car, and bicycle. Results for each type of object are individually counted by an operator, as shown in Table V, where "total" is the sum of the aforementioned three counters. Considering that either "perfect" or "broken" contributes to successful tracks, although some of them might be partial ones, the tracking ratio "t.ratio" is calculated as "(perfect+broken)/total." Considering that only a perfect track provides a full path of the moving object from its entrance to the exit of the intersection, perfect ratio "p.ratio" is calculated as "perfect/total." Similar to the results for detection, the tracking results of cars and bicycles are quite good. As for pedestrians,

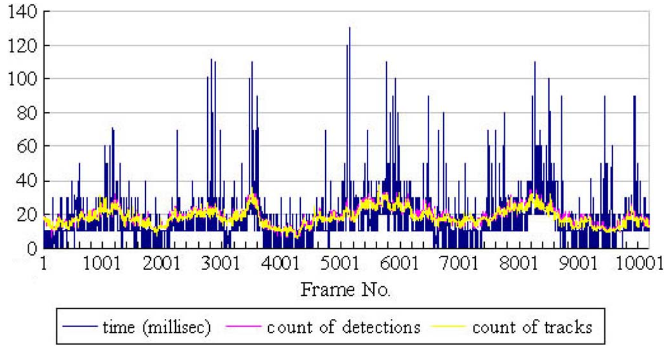


Fig. 15. Counts of detections and tracks at each frame, as well as the computation cost.

the tracking ratio is lower than the detection ratio, as some of the merged detections (counted in the detection ratio) result in error tracks. However, a perfect ratio is improved in the tracking result, from 0.551 to 0.811, meaning that some of the imperfect detections do not affect the final tracking results or that they are compensated for through the tracking procedure. Nevertheless, the perfect ratio in tracking pedestrians is the lowest; their tracks are easily merged or hijacked, particularly in cluttered situations.

3) Computational Cost: All objects inside the measurement range of the laser scans are detected and tracked (not only within the center zone of the intersection). Their counts at each frame are shown in Fig. 15. The count of the tracks is slightly different for detections; however, this discrepancy is not obvious in Fig. 15. The counts at each frame are within a range of 7–35 (an average is estimated to be 18), reflecting the usual capacity for an intersection. In addition, the computational cost (time) at each frame is shown in Fig. 15. It is obvious that the computational cost is related to the number of objects; both change in a synchronized fashion. The computational cost at each frame is mainly within the range of [0–60] ms, with only a few frames higher than this. The average time is calculated to be 14.5 ms. Although the accuracy in the time measurement is limited to within the error range of the Windows application programming interface, compared to the frame rate for data processing, which is 10 Hz (100 ms/frame), such a computational cost should allow for online processing.

D. Discussions and Future Works

In addition to the imperfect ratios in the detection and tracking of pedestrians, there are still problems that need to be improved through future work. We omit those problems that might be caused by software implementation and instead discuss the limitations of the algorithms and those limitations that lead to future work. Fig. 16 provides more information.

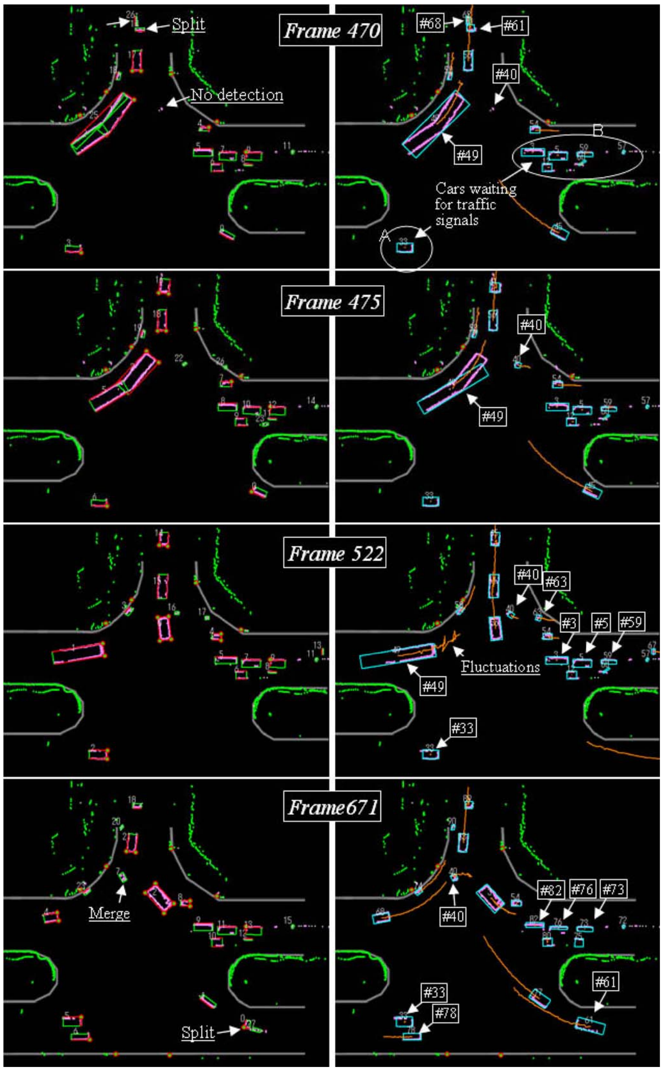


Fig. 16. Negative results to study the algorithm limitations.

1) Two-car buses: Some of the buses in Beijing are made of two cars, and the shapes of their bodies change when they make turns. This construction is a big challenge to our method, as we assume a rigid body in our object model, i.e., a rectangle, for all moving objects. From frames #470 to #522, a right-turning bus #49 is detected and tracked. It can be found from the detection results that the number of detections varies as the bus changes its shape, e.g., at frame #470, there are three detections; at frame #475, there are two detections; and at frame #522, there is one detection. Such unreliable detections yield large errors in state estimations, resulting in a largely fluctuating trajectory, as shown at frame #522. It is important to extend our definitions of object models to improve on the detection and tracking results of two-car buses.
2) Unreliable results at an entrance zone: At the point where objects are far from laser scanners, detection results might be unstable because of the lower scan resolution and/or occlusions. This scenario can somehow be recovered through tracking. For example, at frame #671, a split happens to the detection of car #61, whereas a successful tracking result is achieved. As its state parameters have already been reliably estimated though tracking during the past frames, the multidetections are merged in the process of solving the association matrix. This happens in tracking most of the moving objects that leave the intersection, while it is still risky to trust the tracking result when the object's measurement is extremely inaccurate. However, for an object entering the scene, the detection result is

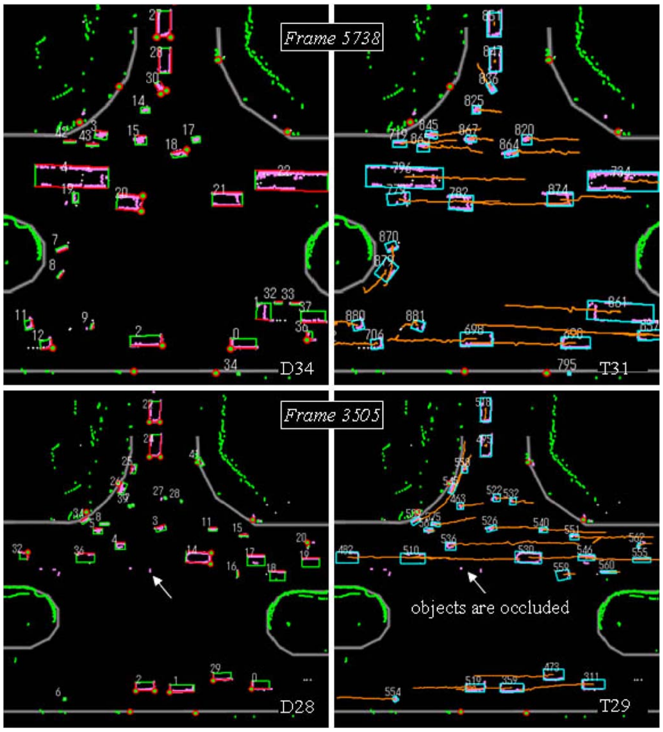


Fig. 17. Results in very dynamic situations.

more critical for tracking. At frame #470, a split happens in detection, and a new track #68 is created in addition to track #61 (see also Fig. 10). Fortunately, track #68 expires after a few frames. However, in some other cases, a new track hijacking the original track could happen. Normally, zones where objects enter the intersection are beyond the center zone of sensor coverage. Unstable detections and unreliable tracking leave many short trajectory fragments or hijacked trajectories in the final output. Future work needs address how to improve the trajectory quality.

3) Temporarily static moving objects: Typical examples are cars waiting for a traffic signal. As described at frame #470, zone A has one waiting car #33 that is clearly viewed from the sensors. At frame #522, three sides of car #33 are observed, with two corner points reliably detected. When car #78 passes by, the right side of car #33 is blocked from measurement; however, the tracking result is not affected, as its two other sides are still reliably measured. This situation is not the same as the situation in zone B, where a number of cars are waiting and each has a great deficit in its measurements. As they are at the entrance to the intersection, the tracking module still has no reliable state estimations for them. Their tracking is easily affected due to the occlusions from other moving objects, e.g., the right-turning cars nearby. These cars can be found from the change of track ID (#3, #5, #59) at frame #522 to (#82, #76, #73) at frame #671. An algorithm that integrates *a priori* knowledge of traffic environments might be a solution to such problems.

4) "Black hole"s: Fig. 17 shows result from very dynamic scenes. Numbers shown in the bottom-right corner of each subfigure are the counts of either detections (denoted by "D") or tracks (denoted by "T"). At frame #5738, there are a total of 34 detections, with 31 moving objects being tracked. It is visually evident from the results that all moving objects at the intersection are correctly tracked. At frame #3505, there are fewer moving objects in the scene. However, because of their spatial distributions, some objects in the middle area of the intersection are almost completely blocked from laser scanners. We call this type of blockage a "black hole." A small black hole will cause a short blockage in the object's measurement (e.g., a detection failure to the object) but would not critically affect the object's tracking. However, if a moving object enters a large black hole, where a continuous detection failure happens, its track expires. When the object is measured again, a track with a new ID is created. Black holes will lower the perfect ratio in the tracking result. Toward a real application at intersections, it is important to develop a simulation system that predicts the system limitations in extreme situations and optimize the sensor layouts. This type of system will be addressed in future work.

VI. Conclusion

This paper has proposed an algorithm for moving-object detection and tracking at intersections using integrated data from a network of horizontal laser scanners. The goal is to detect all moving objects that enter the intersection, estimate their state parameters, and track their motion trajectories during their movement through the intersection. To accomplish this goal, an object model has been defined using the special characteristics of laser scanning so that the partial observations of each individual side of the moving object can be discriminated and reliability items can be associated with feature parameters to discriminate direct observations from occluded features; a detection algorithm has been developed by focusing on grouping the measurements of the same moving objects from different laser scanners at a single data frame; a tracking algorithm addresses the association of data measurements along temporal axis, so that information about different sides of the object can be integrated along time; and a more accurate estimation to object state can be achieved. Algorithms are tested using the real intersection data collected from experiments in central Beijing through both case studies and quantitative evaluations. Promising results are demonstrated, and limitations and future work are also discussed.

Appendix A
Definitions of the Parameters in an Object Model

1) v_i and rv_i: For each directional vector v_i, we define a reliability item rv_i. If a directional vector u is extracted from a data cluster that is associated with an object and a match is made between u and v_i, then we say that side v_i is supported by an observation u and set $rv_i = true$. u is called a support vector of side v_i.

2) c_i and rc_i: For each corner point c_i, we define a reliability item rc_i. If two neighboring sides v_i and v_j are both supported, i.e., $rv_i \& rv_j$, where $j = (i+1) mod\, 4$, their point of intersection is assigned to corner point c_i, and the corresponding reliability is set to $rc_i = true$.
3) l_i and rl_i: For each dimensional size l_i, we define a reliability item rl_i. If two neighboring corner points c_k and c_j are both valid, i.e., $rc_k \& rc_j$, where $j = (k+1) mod\, 4$, meaning that a full-dimensional size of the side is supported by two reliably extracted corner points, the Euclidean distance between them is assigned to l_i, where $i = 1$ if $k = 2, 4$, or $i = 2$ if $k = 1, 3$, and subsequently, $rl_i = true$. Similarly, if a pair of opposite sides v_k and v_j are both supported, i.e., $rv_k \& rv_j$, where $j = (k+2) mod\, 4$, the orthogonal distance between them is assigned to l_i, where $i = 1$ if $k = 2, 4$, or $i = 2$ if $k = 1, 3$, and $rl_i = true$.
4) p and rp: For center point p, we define a reliability item rp. Center point p is a latent variable of the object, which can only be estimated from other observable feature parameters. For example, if corner point c_1 is valid, we can estimate a center point as $p = c_1 - 0.5 * l_1 * v_1 + 0.5 * l_2 * v_2$. Thus, reliability rp is evaluated on referencing feature parameters. Here, we define $rp = true$ if and only if at least one corner point is valid, at least one directional vector is supported, and both dimensional sizes are full observations.

Appendix B
Estimating an Object Model on a Set of Clusters

Given a set $\{c_j\}$ of one or more clusters, estimating an object model m of a detection g_j is to fit parameters that are defined in Table I. Each cluster c_j is a composition of (u_j, L_j, P_j), where u_j is the directional vector of an observed edge, L_j is the length, and P_j is the set of laser points on the edge. The parameters of m are sequentially estimated here.

1) *Directional vectors* v_i: Assign v_1 to the u_j of the longest L_j. If no valid u_j exists, set v_1 to default values. Assign v_2, v_3, and v_4 by rotating v_1 counterclockwise to 90^o, 180^o, and 270^o, respectively.
2) *Support of the directional vectors* rv_i: Match each u_j with $v_{1,\ldots,4}$. If there is a match between u_j and v_i, then u_j is said to be a support vector of v_i, and set $rv_i = true$; the set of laser points P_j are associated with the side of v_i. If a u_j cannot be matched with any of the v_i, and its length L_j is longer than a predefined threshold, then an observed edge that has high reliability does not match with the model, and the procedure ended in failure.
3) *Corner points* c_i *and reliabilities* rc_i: If a pair of neighboring sides are both supported, assign their intersection points to the corresponding corner point c_i, and set $rc_i = true$.
4) *Dimensions* l_i *and reliabilities* rl_i: If a pair of neighboring corner points are both valid, then assign their Euclidean distance to the corresponding l_i, and set $rl_i = true$. If a pair of opposite sides is both supported, then assign their orthogonal distance to the corresponding l_i, and set $rl_i = true$. After the preceding procedures, if a $rl_i = false$, e.g., $i = 1$, then project the laser points that are associated to both v_1 and v_3, and assign l_1 to their distribution dimension along v_1. It is the same if $i = 2$. The l_is are finally validated with the knowledge of the size of a normal moving object. For example, if the shorter l_i is larger than a lane width or the longer l_i is longer than a normal bus, implying that a merge of multiple moving objects might happen, the procedure ended in failure.
5) *Center point p and reliability rp*: If a corner point is valid, estimate a center point on other feature parameters, e.g., $rc_1 = true$, estimate $p = c_1 - 0.5 * l_1 * v_1 + 0.5 * l_2 * v_2$. Otherwise, find a center point of the laser points. If there exists a valid corner point and a support vector and both dimensions are valid, then set $r_p = true$; otherwise, $r_p = false$.
6) The procedure ended in success.

Appendix C
Hypothesis Generation in Case SS3–SS5

In case SS3, the measurement data of g^k are a small point cloud, whereas t^{k-1} suggests a moving object with a large dimension. If g^k is associated with t^{k-1}, then g^k might be an observation of a partial side of the object. A large number of hypotheses can be generated in such a case, whereas for the sake of computational cost, we predict the location of a center point at the next frame only and other state parameters remain the same. In the example of Fig. 9, if tv_3 is matched with center point gp, then tp is projected to point C. Similarly, we can decide on a rectangle ABCD by representing a prediction space at center point tp. Thus, in SS3, the hypotheses are generated by taking equal prediction samples for the center point in a rectangular area, which are considered to have the lowest reliability because of extreme occlusion.

In case SS4, both t^{k-1} and g^k are of small dimensions. A prediction sample can be generated by projecting center point tp to gp. Thus, in SS4, a single hypothesis is generated and considered to have high reliability.

In case SS5, t^{k-1} suggests a small object, whereas g^k is a measurement of a large dimension, which could be fitted onto a rectangular shape. This scenario could happen when a car moves from far away to a central area, with the measurement data obtained starting at poor quality and becoming high quality, so that, in case SS5, we use the object model of g^k to generate a single hypothesis for t^{k-1}. However, as there are dramatic changes in state parameters, the lowest reliability of the hypothesis is considered.

References

[1] M. Valera1 and S. A. Velastin, "Intelligent distributed surveillance systems: A review," *Proc. Inst. Elect. Eng.—Vis. Image Process.*, vol. 152, no. 2, pp. 192–204, Apr. 2005.

[2] D. B. Reid, "A multiple hypothesis filter for tracking multiple targets in a cluttered environment," Lockheed Missiles Space Co., Sunnyvale, CA, Tech. Rep. LMSC, D-560254, 1977.

[3] D. B. Reid, "An algorithm for tracking multiple targets," *IEEE Trans. Autom. Control*, vol. AC-24, no. 6, pp. 843–854, Dec. 1979.

[4] C. J. Pai, H. R. Tyan, Y. M. Liang, H. Y. Liao, and S. W. Chen, "Pedestrian detection and tracking at crossroads," *Pattern Recognit.*, vol. 37, no. 5, pp. 1025–1034, May 2004.

[5] R. Rad and M. Jamzad, "Real time classification and tracking of multiple vehicles in highways," *Pattern Recognit. Lett.*, vol. 26, no. 10, pp. 1597–1607, Jul. 2005.

[6] Z. Kim, "Realtime object tracking based on dynamic feature grouping with background subtraction," *Proc. IEEE Comput. Vis. Pattern Recog.*, pp. 1–8, 2008.

[7] S. Kamijo, Y. Matsushita, K. Ikeuchi, and M. Sakauchi, "Traffic monitoring and accident detection at intersections," *IEEE Trans. Intell. Transp. Syst.*, vol. 1, no. 2, pp. 108–118, Jun. 2000.

[8] H. Veeraraghavan, O. Masoud, and N. Papanikolopoulos, "Computer vision algorithms for intersection monitoring," *IEEE Trans. Intell. Transp. Syst.*, vol. 4, no. 2, pp. 78–89, Jun. 2003.

[9] S. Atev, H. Arumugam, O. Masoud, and R. Janardan, "A vision-based approach to collision prediction at traffic intersections," *IEEE Trans. Intell. Transp. Syst.*, vol. 6, no. 4, pp. 416–423, Dec. 2005.

[10] Z. Tian, M. Kyte, and H. Liu, "Vehicle tracking and speed measurement at intersections using video-detection systems," *ITE J.*, vol. 79, no. 1, pp. 42–46, 2009.

[11] S. E. Shladover, Z. Kim, M. Cao, A. Sharafsaleh, and J.-Q. Li, "Bicyclist intersection crossing times: Quantitative measurements for selecting signal timing," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2128, pp. 86–95, 2009.

[12] T. Zhao and R. Nevatia, "Car detection in low resolution aerial image," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2001, vol. 1, pp. 710–717.

[13] I. Cohen and G. Medioni, "Detecting and tracking moving objects for video surveillance," *Proc. IEEE Comput. Vis. Pattern Recog.*, vol. 2, pp. 319–325, 1999.

[14] D. M. Gavrila and S. Munder, "Multi-cue pedestrian detection and tracking from a moving vehicle," *Int. J. Comput. Vis.*, vol. 73, no. 1, pp. 41–59, Jun. 2007.

[15] M. Betke, E. Haritauglu, and L. S. Davis, "Real-time multiple vehicle detection and tracking from a moving vehicle," *Mach. Vis. Appl.*, vol. 12, no. 2, pp. 69–83, Aug. 2000.

[16] Z. Sun, G. Bebis, and R. Miller, "On-road vehicle detection: A review," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 5, pp. 694–711, May 2006.

[17] B. Leibe, K. Schindler, N. Cornelis, and L. V. Gool, "Coupled object detection and tracking from static cameras and moving vehicles," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 10, pp. 1683–1698, Oct. 2008.

[18] D. Streller, K. Dietmayer, and J. Sparbert, "Vehicle and object models for robust tracking in traffic scenes using laser range images," in *Proc. IEEE Int. Conf. Intell. Transp. Syst.*, 2001, pp. 118–123.

[19] E. Prassler, J. Scholz, and A. Elfes, "Tracking people in a railway station during rush-hour," in *Proc. ICVS*, vol. 1542, *LNCS*, H. I. Christensen, Ed., 1999, pp. 162–179.

[20] A. Mendes and U. Nunes, "Situation-based multitarget detection and tracking with laserscanner in outdoor semi-structured environment," in *Proc. IEEE/RSJ Conf. Intell. Robots Syst.*, 2004, vol. 1, pp. 88–93.

[21] T. Vu, O. Aycard, and N. Appenrodt, "Online localization and mapping with moving object tracking in dynamic outdoor environment," in *Proc. IEEE Intell. Veh. Symp.*, 2007, pp. 190–195.

[22] F. Fayad and V. Cherfaoui, "Tracking objects using a laser scanner in driving situation based on modeling target shape," in *Proc. IEEE Intell. Veh. Symp.*, 2007, pp. 44–49.

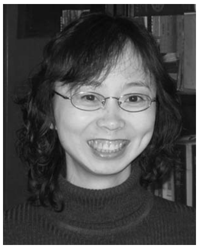
[23] C. C. Wang, C. Thorpe, S. Thrun, M. Hebert, and H. Durrant-Whyte, "Simultaneous localization, mapping and moving object tracking," *Int. J. Robot. Res.*, vol. 26, no. 9, pp. 889–916, Sep. 2007.

[24] L. Alexander, P. M. Cheng, A. Gorjestani, A. Menon, B. Newstrom, and C. Shankwitz, "The minnesota mobile intersection surveillance system," in *Proc. IEEE Intell. Transp. Syst. Conf.*, 2006, pp. 139–144.

[25] H. Zhao and R. Shibasaki, "A novel system for tracking pedestrians using multiple single-row laser range scanners," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 35, no. 2, pp. 283–291, Mar. 2005.

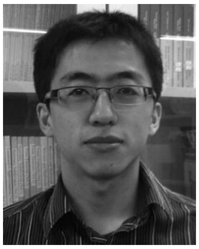
[26] K. Nakamura, H. Zhao, R. Shibasaki, K. Sakamoto, T. Ohga, and N. Suzukawa, "Tracking pedestrians using multiple single-row laser range scanners and its reliability evaluation," *Syst. Comput. Jpn.*, vol. 37, no. 7, pp. 1–11, Jun. 2006.

[27] H. Zhao, J. Cui, H. Zha, K. Katabira, X. Shao, and R. Shibasaki, "Sensing an intersection using a network of laser scanners and video cameras," *IEEE Intell. Transp. Syst. Mag.*, vol. 1, no. 2, pp. 31–37, Summer 2009.



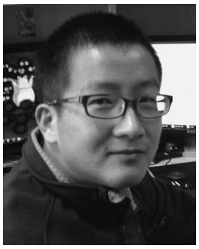
Huijing Zhao (M'07) received the B.S. degree in computer science from Peking University, Beijing, China, in 1991 and the M.E. and Ph.D. degrees in civil engineering from the University of Tokyo, Tokyo, Japan, in 1996 and 1999, respectively.

From 1991 to 1994, she was with Peking University, where she was involved with a project developing a graphic information system platform. In 2003, after several years of postdoctoral research with the University of Tokyo, she became a Visiting Associate Professor with the Center for Spatial Information Science. In 2007, she joined Peking University as a Professor with the Key Laboratory of Machine Perception (MOE), and the School of Electronics Engineering and Computer Science. Her research interests include machine perception, intelligent vehicles, and spatial data handling.



Jie Sha received the B.S. degree in automation from Tsinghua University, Beijing, China, in 2008 and the M.S. degree in machine intelligence from Peking University, Beijing, in 2011. His M.S. study focused on collecting and modeling the motion trajectories of moving objects at an intersection for behavior analysis.

His research interests include computer vision and machine perception.



Yipu Zhao received the B.S. degree in machine intelligence in 2010 from Peking University, Beijing, China, where he is currently working toward the M.S. degree with the Department of Machine Intelligence, School of Electronics Engineering and Computer Science. His M.S. study is focused on object discovery and modeling from the laser scan data of an urban scene.

His research interests include computer vision and machine perception.

Junqiang Xi received the B.S. degree in automotive engineering from Harbin Institute of Technology, Harbin, China, in 1995 and the Ph.D. degree in vehicle engineering from Beijing Institute of Technology (BIT), Beijing, China, in 2001.

In 2001, he joined the State Key Laboratory of Vehicle Transmission, BIT, and in 2003, he was promoted to Associate Professor. His research interests include intelligent transportation system, mechanics, and vehicular engineering.

Jinshi Cui (M'10) received the B.S. and Ph.D. degrees in computer science from Tsinghua University, Beijing, China, in 1999 and 2004, respectively.

In 2004, she joined the School of Electronics Engineering and Computer Science, Peking University, Beijing, as an Assistant Professor. She was promoted to Associate Professor in 2007. Her research interests include computer vision and robotics.

Hongbin Zha (M'06) received the B.E. degree in electrical engineering from Hefei University of Technology, Hefei, China, in 1983 and the M.S. and Ph.D. degrees in electrical engineering from Kyushu University, Fukuoka, Japan, in 1987 and 1990, respectively.

After being a Research Associate with Kyushu Institute of Technology, he joined Kyushu University in 1991 as an Associate Professor. In 1999, he was also a Visiting Professor with the Centre for Vision, Speech, and Signal Processing, Surrey University, Surrey, U.K. Since 2000, he has been a Professor with the Center for Information Science, School of Electronics Engineering and Computer Science, Peking University, Beijing, China. He has published more than 200 technical publications in journals, books, and international conference proceedings. His research interests include computer vision, digital geometry processing, and robotics.

Dr. Zha was the recipient of the Franklin V. Taylor Award from the IEEE Systems, Man, and Cybernetics Society in 1999.

Ryosuke Shibasaki (M'10) received the B.S., M.S., and Ph.D. degrees in civil engineering from the University of Tokyo, Tokyo, Japan, in 1980, 1982, and 1987, respectively.

From 1982 to 1988, he was with the Public Works Research Institute, Ministry of Construction. From 1988 to 1991, he was an Associate Professor with the Department of Civil Engineering, University of Tokyo. In 1991, he moved to the Institute of Industrial Science of the same university. In 1998, he was promoted to Professor with the Center for Spatial Information Science, University of Tokyo. His research interests include 3-D data acquisition for Geographic Information Systems (GIS), conceptual modeling for spatial objects, and agent-based micro simulation in a GIS environment.