A Method for Vehicle Count in the Presence of Multiple-Vehicle Occlusions in Traffic Images

Clement Chun Cheong Pang, *Member, IEEE*, William Wai Leung Lam, and Nelson Hon Ching Yung, *Senior Member, IEEE*

***Abstract*—This paper proposes a novel method for accurately counting the number of vehicles that are involved in multiple-vehicle occlusions, based on the resolvability of each occluded vehicle, as seen in a monocular traffic image sequence. Assuming that the occluded vehicles are segmented from the road background by a previously proposed vehicle segmentation method and that a deformable model is geometrically fitted onto the occluded vehicles, the proposed method first deduces the number of vertices per individual vehicle from the camera configuration. Second, a contour description model is utilized to describe the direction of the contour segments with respect to its vanishing points, from which individual contour description and vehicle count are determined. Third, it assigns a resolvability index to each occluded vehicle based on a resolvability model, from which each occluded vehicle model is resolved and the vehicle dimension is measured. The proposed method has been tested on 267 sets of real-world monocular traffic images containing 3074 vehicles with multiple-vehicle occlusions and is found to be 100% accurate in calculating vehicle count, in comparison with human inspection. By comparing the estimated dimensions of the resolved generalized deformable model of the vehicle with the actual dimensions published by the manufacturers, the root-mean-square error for width, length, and height estimations are found to be 48, 279, and 76 mm, respectively.**

***Index Terms*—Deformable model, occlusion, occlusion reasoning, resolvability, vehicle counting, vehicle segmentation, vehicle tracking.**

I. Introduction

TRACKING of moving vehicles in traffic images is a challenging research topic as it has to directly deal with hostile but realistic conditions on the road, such as uncontrolled illuminations, cast shadows, and visual occlusion [1]. Yet, the outcome of being able to accurately count and resolve vehicles under such conditions has tremendous benefit to traffic surveillance. Accurate vehicle count enables the extraction of important traffic information such as congestion level and lane occupancy, while accurate resolution of vehicles provides essential input for vehicle tracking, which can result in semantic traffic information, such as travel time estimation and driving behavior analysis. Such information can be disseminated to road users, which can potentially reduce traffic congestion and environmental pollution as a consequence, and enhance road safety [2], [3]. Of all the problems that are associated with outdoor tracking, visual occlusion presents the biggest obstacle. It appears frequently in urban and congested traffic [4]–[10], and introduces ambiguity in vehicle counting, which naturally depletes our ability and accuracy in tracking these vehicles [3], [11], [12]. It further leads to erratic estimation of traffic volume, flow, and other traffic parameters. Although a partial solution may be obtained by setting the camera's optical axis perpendicular to the road plane, this configuration severely reduces the range and amount of visual information available, as compared with a perspective camera configuration [13]. An alternative solution is to use more than one camera (i.e., stereo vision) [14] or use nonvisual sensors such as inductive loops or laser sensors in addition to the visual sensors to aid the vision system [15], [16]. However, it is generally known that stereo vision makes little difference from monocular vision under congested situations, and the added installation and signal-processing complexities of nonvisual sensors simply depletes its usability [3]. Not surprisingly, most current research in occlusion handling assumed a perspective monocular configuration for detecting minor to moderate occlusion problems involving two or three vehicles in noncongested highways [17]–[20]. These works represent an important first step in solving the visual occlusion problem, but clearly, a more general approach for resolving severe multiple-vehicle occlusions is desirable. It is the purpose of this paper to propose one such method that can accurately count the number of vehicles that are involved in occlusion and resolve these vehicles into individual vehicles under different level of occlusion severity.

The rest of this paper is organized as follows: a brief problem analysis is given in Section II, which is followed by an overview of the recent related research in this area in Section III. The proposed method is detailed in Section IV. Test results and discussions are presented in Section V. Suggestions for future development are given in Section VI, and concluding remarks can be found in Section VII.

II. Problem Analysis

In this paper, occlusion refers to the visual obstruction of vehicles from one another in an image due to the perspective view of the camera [21]. When we consider this problem, we observe that first, vehicles that are involved in the occlusion are not uniform in size [Fig. 1(a)]. Second, the severity of

Manuscript received June 13, 2005; revised November 29, 2005, May 17, 2006, and August 19, 2006; accepted August 31, 2006. This work was supported in part by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China, under Project HKU7198/05E and in part by the Postgraduate Studentship of the University of Hong Kong. The Associate Editor for this paper was N. Zheng.

The authors are with the Laboratory for Intelligent Transportation Systems Research, Department of Electrical and Electronic Engineering, University of Hong Kong, Hong Kong, China (e-mail: ccpang@eee.hku.hk; wllam@eee.hku.hk; nyung@eee.hku.hk).

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.2007.902647

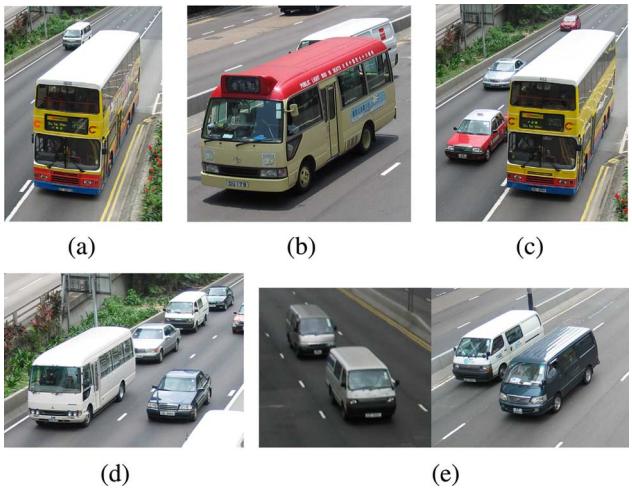


Fig. 1. Vehicle occlusion cases. (a) Different vehicle sizes, (b) severe occlusion, (c) vehicle cluster, (d) vehicle queue, and (e) PTZ views.

occlusion varies according to the spatial relationship between vehicles, i.e., a vehicle might be totally occluded by one or more vehicles, unoccluded, or partially occluded [Fig. 1(b)]. Third, the number of vehicles involved can be up to N vehicles in a cluster [Fig. 1(c)] or in a queue [Fig. 1(d)]. Fourth, the camera viewing configuration changes when the camera undergoes pan–tilt–zoom (PTZ) action [Fig. 1(e)].

From our first observation, it can be deduced that vehicle dimension provides no strong clues as to how severe the occlusion is. Therefore, a good detection method should not rely on vehicle size. From our second observation, both the totally occluded and unoccluded vehicles require no action, whereas the partially occluded vehicles are our major concern. Although over time, the severity of occlusion may change, it would be logical to stay focused on resolving the occlusion problem in each image before extending it to the time axis. From our third observation, our solution must be general enough to solve the occlusion of N vehicles in a cluster or queue. From our fourth observation, as PTZ actions vary the camera parameters as well as the number of contour outlines and vertices that are seen by the camera, it is important that the proposed method formalizes such relationship between the camera configuration and the vehicle model.

III. Related Works

Research in occlusion detection/handling has been considered as the key to practical vehicle or human tracking [11]. Many published solutions to the problem of human occlusion exploited the uniformity and uniqueness of human shape [22], while those for vehicle occlusion evolved into a variety of directions because of shape diversity. In general, solutions to both human and vehicle problems extend from spatial to temporal domains and can be broadly classified as follows: model-based [23], region-based [19], active contour-based [17], feature-based [24], stereo vision-based [14], and probabilistic-based [22]. Hybrid approaches [25] will not be separately discussed here.

The basic principle of model-based methods is to define a 3-D geometric model as a matching template that describes the shape of a vehicle. Instead of defining a large number of rigid models, a single parameterized or deformable model is often adopted for all vehicle types [26]. In most cases, it works well in free-flow traffic and can be readily adapted to deal with background occlusion. For instance, Yang *et al.* [23] proposed a method that claimed to work under severe background occlusion. Unfortunately, all they have tested were toy cars in a laboratory environment. A more rigorous and stringent set of roadside tests would be desirable for verification. Ikeda *et al.* [18] described a more realistic approach and achieved 94.8% of count accuracy for moderate occlusion of two vehicles. However, in their method, only the front or back of the vehicles can be identified; thus, the method cannot resolve all 3-Ds of the vehicle.

The idea of region-based tracking is to connect regions as "blobs" to represent vehicles. It works well when vehicles are unoccluded; otherwise, segmenting larger blobs into groups of smaller ones can be challenging. To deal with this, a spatio–temporal Markov random field model was proposed to work on the texture correlation of predefined subimages and segment the subimages into most likely objects [19]. It achieved a count accuracy of 94.6%, for moderate occlusion that involves two or three vehicles only, which is similar to the results that were obtained by Ikeda *et al.* [18]. Other tools such as Kalman filtering and graph optimization have also been proposed [20], resulting in a blob relationship graph to represent the split and merge of the blobs between each subsequent frame, with even lower count accuracy of 85%.

Similarly, active contour-based methods used active contours to model the boundary of vehicles, which can be updated dynamically. They have relatively lower computational complexity, but contour initialization for each vehicle is a complex issue [17], which is difficult to achieve efficiently in practice. In some cases, shape restriction is applied together with a Kalman filter to estimate the spatio–temporal relationships of the contour to improve tracking robustness in the presence of occlusion [27]. Experimental results were encouraging for tracking vehicles that are well-separated but not conclusive in terms of their ability to resolve occlusion effectively since results that contain vehicle occlusion were not presented in their work.

On the other hand, feature-based methods track vehicle subfeatures instead of the entire vehicle. The assumption is that, even in the presence of partial occlusion, subfeatures of vehicles may still be visible. The question is on how to collate these subfeatures in the occluded cases. Gentile *et al.* [24] proposed using a Kalman filter, with the condition that less than 30% of these subfeatures are occluded. In practice, this is seldom the case.

In stereo-based methods, two or more images of the same scene are used to track densely populated objects. Otsuka and Mukawa [14] modeled the spatial structure of the occlusion process between human and its uncertainty, and formulated a recursive Bayesian estimation method for human position and posture. Although they have tested their model using six cameras on five people successfully, its reliance on a large number of camera views implies high computational complexity, which could be a hindrance to vehicle tracking.

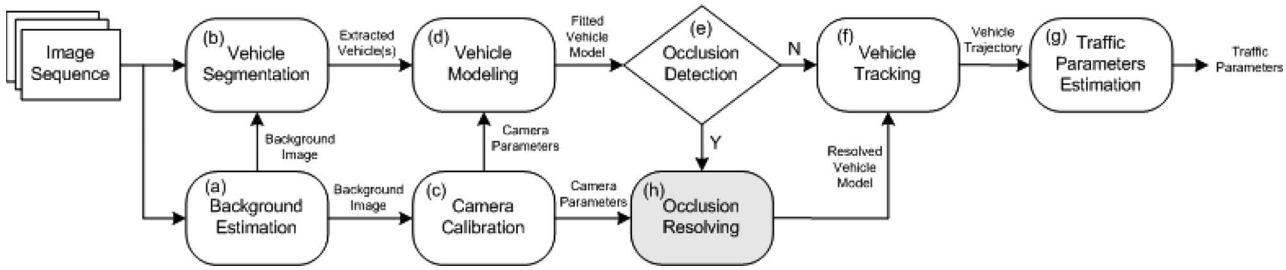


Fig. 2. Conceptual diagram of a VTS system.

In probabilistic-based methods, a Bayesian model is often used for segmentation and tracking. Eng *et al.* [22] proposed a human shape model for foreground validation and occlusion handling, while segmentation is obtained when a maximum *a posteriori* value is found. Their results have demonstrated that it can resolve slight occlusion of two humans. Applying this to vehicle occlusion is yet to be studied as shape is not as uniform, while occlusion can be more severe when more vehicles are involved, and vehicle positions are not known *a priori*.

In summary, we have observed that there is a common problem among existing occlusion handling methods, i.e., they are designed for resolving the occlusion of two or three vehicle/human only [18]–[20], [22]. Although [14] has attempted to deal with the occlusion of five people, the problem of N-vehicle/human occlusion has still not been tackled. Moreover, evidence has shown that, if occlusion is too severe, most of these methods would fail [23], [24]. Therefore, we propose a generalized deformable model (GDM)-based method to deal with the occlusion of N vehicles, which is characterized by its simplicity and generality.

IV. Proposed Method

A. Overview

A visual traffic surveillance (VTS) system can be defined as depicted in Fig. 2 [3]. In the figure, background estimation [Fig. 2(a)] estimates the background from the image. The "scoreboard algorithm" from [28] is employed in this paper for this purpose, as it is fast and accurate for estimating the stationary background. Vehicle segmentation [Fig. 2(b)] segments the vehicles from the background. The texture-based vehicle segmentation approach that is described in [29] is employed to extract the moving vehicles since it can do that without being affected by vehicle cast shadow. Camera calibration [Fig. 2(c)] determines the camera parameters from the background image. The camera calibration method that was proposed by [30] is employed to determine the camera parameters for its ability to cope with all camera PTZ actions. Vehicle modeling [Fig. 2(d)] fits the GDM onto the binary vehicle mask by transforming it from 3-D world coordinates to 2-D image coordinates based on the camera parameters. The detail of the method that was used can be found in [31]. Occlusion detection [Fig. 2(e)] determines whether there is occlusion inside the fitted GDM. The area ratio method that was proposed in [12] is used for occlusion detection. Vehicle tracking [Fig. 2(f)] and traffic parameters estimation [Fig. 2(g)] track the fitted model in the subsequent frames to generate the trajectory of the vehicle,

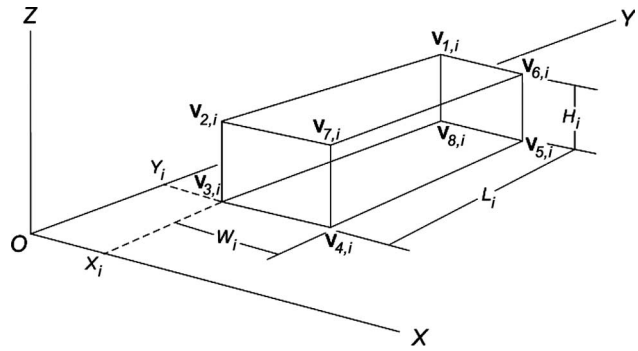


Fig. 3. GDM of vehicle i.

and from the trajectory, traffic parameters such as travel time, speed, driving behavior of individual vehicle, etc. [3] can be estimated. Occlusion resolution [Fig. 2(h)] partitions the GDM into smaller ones such that they have one-to-one relationships with the vehicles [11]. In this paper, our focus is on proposing a method for resolving multiple occluded vehicles and counting the vehicle number, and we assume that the input is a composite vehicle mask of the occluded vehicles [21].

The basic concept of our approach is to count the number of occluded vehicles by evaluating the property of the contour segments of the occluded cluster and to assign a "resolvability index" to each vehicle, which indicates whether the vehicle can be resolved from an occluded scenario, where "1" indicates that the vehicle is resolvable and "0" indicates that it is not resolvable. It begins by assuming that a single GDM can be fitted into N occluded vehicles. Essentially, the GDM is a 3-D rectangular model with 8 vertices (Fig. 3) [3], [31]. The advantage of this model is that it is general enough to represent the shape of all kinds of vehicles.

The block diagram of the proposed occlusion-resolving method is depicted in Fig. 4 [21]. Three predefined models are used to model the aforementioned approaches: 1) the Vertex-Shape Model (VSM) [Fig. 4(a)]; 2) the Contour Description Model (CDM) [Fig. 4(b)]; and 3) the Resolvability Model (RM) [Fig. 4(c)], as shown in the shaded blocks in Fig. 4. Essentially, the purpose of the VSM is to establish a relationship between the camera viewing position and the number of vertices of the projected GDM, as seen in the image. The purpose of the CDM is to model the direction of the lines as well as the shape of the projected GDM from the number of vertices that were defined by the VSM. The RM evaluates whether an occluded GDM is resolvable. Having defined these three models, the number of vehicles N that are involved in an occluded cluster is deduced by first approximating the contour of the composite binary

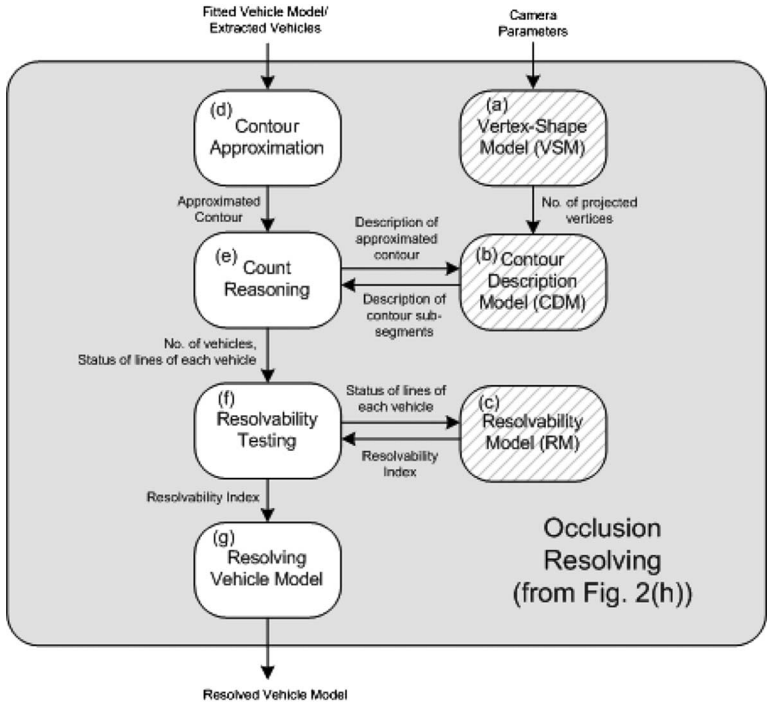


Fig. 4. Proposed method for resolving multiple-vehicle occlusion.

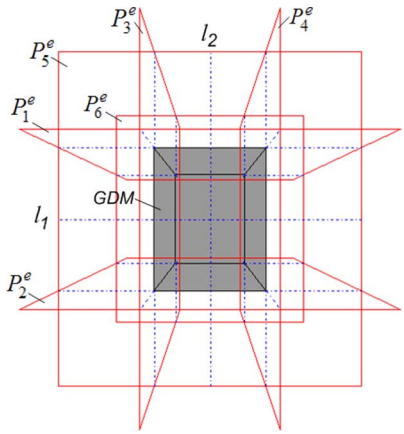


Fig. 5. Virtual extended planes P_i^e.

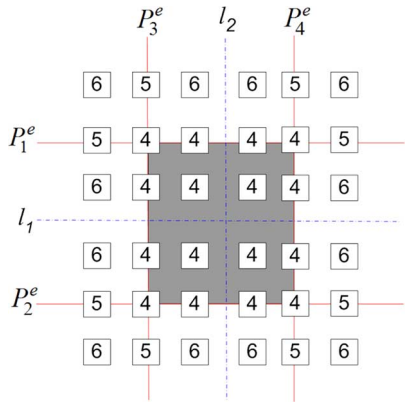


Fig. 6. Relationship between n_v and $\mathbf{Q_c}$.

vehicle mask by the curvature points that were detected on the contour [Fig. 4(d)] and then generating a formal linguistic description of it based on the vanishing points of the road, which is then subdivided into subsegments with reference to the CDM [Fig. 4(e)]. Then, the resolvability index R of each vehicle is determined with reference to the RM, according to the extent of occlusion of the contour lines [Fig. 4(f)]. If the resolvability index is positive (which means that the original GDM of the vehicle is recoverable), then the GDM of the vehicle is reconstructed based on the unoccluded lines of the GDM [Fig. 4(g)] [11]. The final outcome is the resolved GDM that describes the vehicle.

B. VSM

The aim of this model is to establish a relationship between the camera viewing position ($\mathbf{Q}_c = [X_c\ Y_c\ Z_c]^T$) and the number of vertices n_v on the GDM contour that is seen on the image. Since each GDM has eight vertices in 3-D and, at most, six vertices in 2-D, n_v is one of 6, 5, or 4. In order to model n_v with respect to $\mathbf{Q_c}$, we introduce the idea of "virtual extended planes P_i^e" on the GDM in 3-D, which are defined as the surfaces of the GDM being extended to $+\infty$ and $-\infty$ in space but excluding the surfaces of the GDM itself (Fig. 5). As a GDM consists of six surfaces in 3-D, there are six P_i^e's in total (denoted by $P_1^e, P_2^e, \ldots, P_6^e$ in Fig. 5). Lines l_1 and l_2 are dividing lines on plane P_5^e, which divide the GDM into four quarters. The relationship between n_v and $\mathbf{Q_c}$ is modeled in terms of these P_i^e's, lines, and the surfaces of the GDM, which is depicted in Fig. 6. From Fig. 6, if the camera is located in the space bounded by four P_i^e's and a surface of the GDM [Fig. 7(a)], or if it is located on one of the P_i^e's and bounded by two other P_i^e's that are separated by the GDM [Fig. 7(b)], or if it is located at the intersection of two P_i^e's [Fig. 7(c)], then $n_v = 4$. If the camera is located on one of the P_i^e's but not bounded by two other P_i^e's that are separated by the GDM,

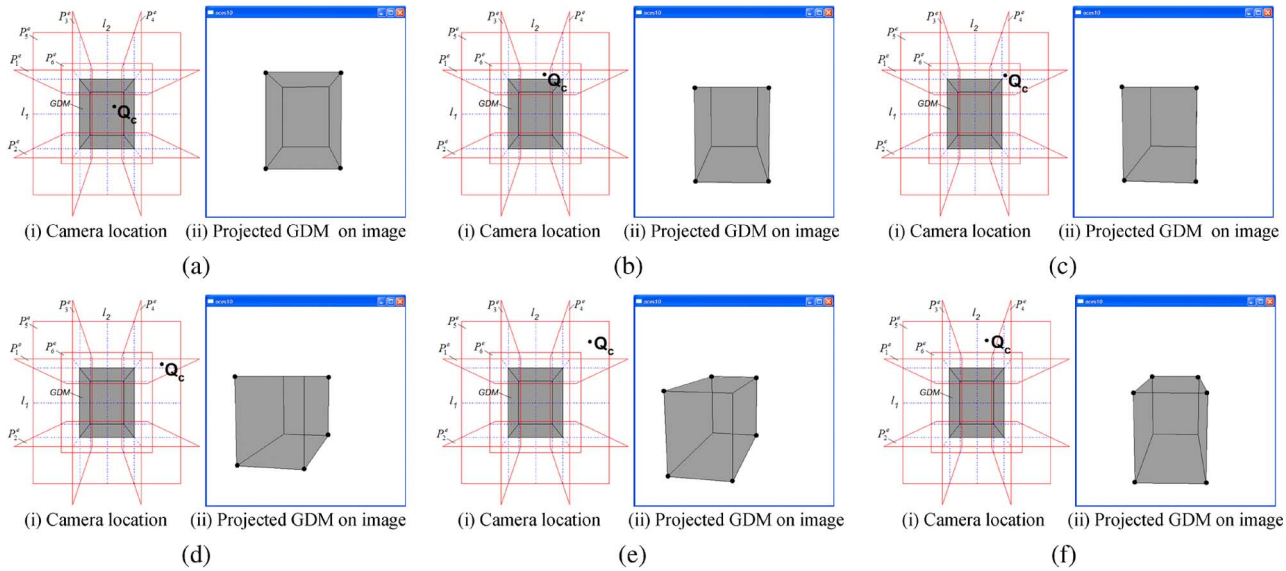


Fig. 7. Number of vertices n_v at different camera locations $\mathbf{Q_c}$. (a) $n_v = 4$. (b) $n_v = 4$. (c) $n_v = 4$. (d) $n_v = 5$. (e) $n_v = 6$. (f) $n_v = 6$.

TABLE I
VSM

$\mathbf{Q}_C$	n_v
Located in the space bounded by 4 P_i^e and a surface of the GDM (Fig. 7(a)) or Located on one of the P_i^e and bounded by two other P_i^e that are separated by the GDM (Fig. 7(b)) or Located at the intersection of two P_i^e (Fig. 7(c))	4
Located on one of the P_i^e but not bounded by two other P_i^e that are separated by the GDM (Fig. 7(d))	5
Otherwise (Fig. 7(e) and (f))	6

then $n_v = 5$ [Fig. 7(d)]. If the camera is located elsewhere, then $n_v = 6$ [Fig. 7(e) and (f)]. The relationship between n_v and $\mathbf{Q_c}$ is summarized in Table I.

C. CDM

The CDM formulates a set of direction primitives to describe the outlines of the projected GDM with respect to the vanishing points $\mathbf{p_x}$, $\mathbf{p_y}$, and $\mathbf{p_z}$ in the image, where $\mathbf{p_y}$ is the vanishing point of the road, $\mathbf{p_x}$ is the vanishing point that is perpendicular to the road direction, and $\mathbf{p_z}$ is the vanishing point that is perpendicular to the ground (Fig. 8). It is assumed that all occluded vehicles share the same set of vanishing points. The direction of the line segments is described in terms of direction primitives "a," "b," "c," "A," "B," and "C" with respect to their relationships with $\mathbf{p_x}$, $\mathbf{p_y}$, and $\mathbf{p_z}$, where "a" denotes a line that intersects $\mathbf{p_y}$ when it is being extended infinitely, and the direction of the line goes *away from* $\mathbf{p_y}$ when the GDM is being traced in a counterclockwise manner (Fig. 9). On the other hand, "A" denotes a line that intersects $\mathbf{p_y}$ when it is extended infinitely but points *toward* $\mathbf{p_y}$ when the GDM is being traced in a counterclockwise manner (Fig. 9). Similarly, "b" and "B" denote the lines that are related to $\mathbf{p_z}$, and "c" and "C" denote the lines that are related to $\mathbf{p_x}$ in the same manner (Fig. 9). If

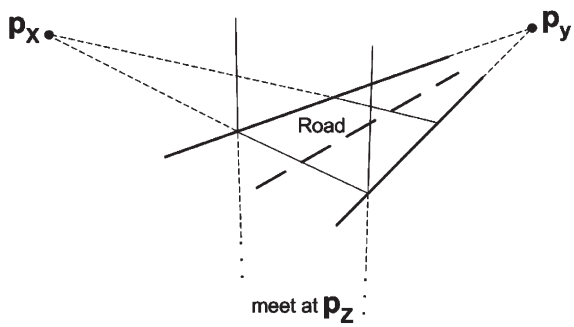


Fig. 8. Vanishing points $\mathbf{p_x}$, $\mathbf{p_y}$, and $\mathbf{p_z}$.

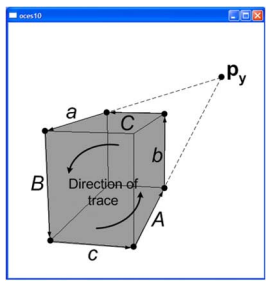


Fig. 9. Description of GDM outline.

a line does not intersect any of $\mathbf{p_x}$, $\mathbf{p_y}$, and $\mathbf{p_z}$ when extended infinitely, then the line is simply rejected and not described by any of the direction primitives. One of the resultant descriptions of the GDM in Fig. 9 is "$aBcAbC$." It should be noted that this description is cyclic, which means that it can also be written as "$AbCaBc$" or "$BcAbCa$," etc.

In some of the cases for $n_v = 4$ or 5, the contour contains a line that intersects more than one vanishing point when the line is being extended infinitely. In such cases, two primitives with an underline are used to represent the line. For example, the GDM in Fig. 10 contains a line that intersects both $\mathbf{p_y}$ and $\mathbf{p_x}$. The underlined convention "$\underline{aC}$" is used to describe the line since the line goes *away from* $\mathbf{p_y}$ first and then goes *toward* $\mathbf{p_x}$ when the contour is traced in a counterclockwise manner. One

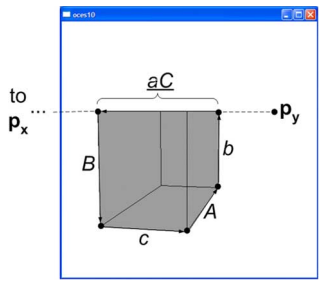


Fig. 10. Case when an extended line intersects two vanishing points.

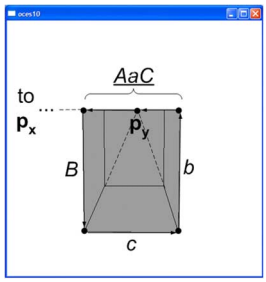


Fig. 11. Case when a vanishing point falls on a line.

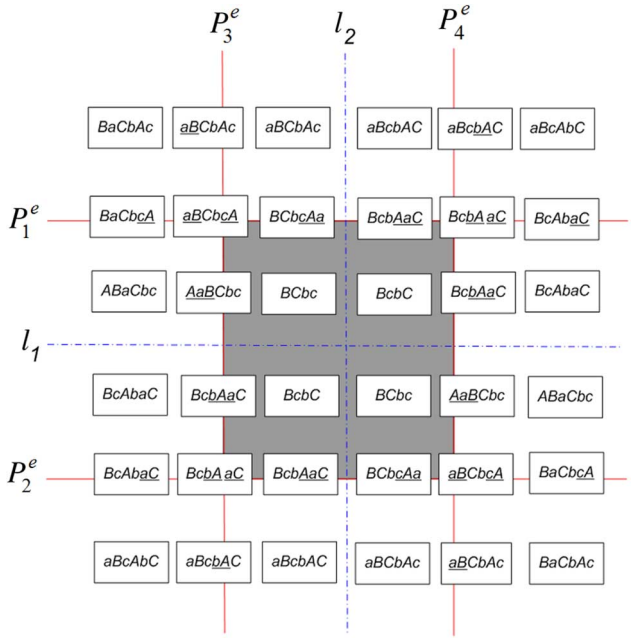


Fig. 12. Description of the GDM outline.

of the resultant descriptions of the GDM is "$BcAba\underline{C}$." In some of the cases for $n_v = 4$, a vanishing point falls on one of the GDM outlines. In such cases, three primitives with an underline are used to represent the line. For example, in Fig. 11, $\mathbf{p_y}$ falls on a GDM outline, so the underlined convention "$\underline{AaC}$" is used to represent the line since the line goes *toward* $\mathbf{p_y}$ first, then goes *away from* $\mathbf{p_y}$, and then goes *toward* $\mathbf{p_x}$ when the contour is traced in a counterclockwise manner. One of the resultant descriptions of the GDM is "$Bcb\underline{AaC}$." The descriptions using direction primitives for all shape types are given in Fig. 12.

To model the shape of the projected GDM, the three cases of $n_v = 4$ in Fig. 7(a)–(c) are modeled by S_4 [Fig. 13(a)]. For the case of $n_v = 5$ in Fig. 7(d), it is modeled by S_5 [Fig. 13(b)].

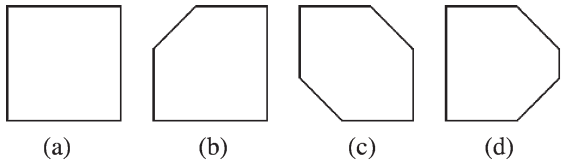


Fig. 13. Shape type of the projected GDM. (a) Type S_4, (b) type S_5, (c) type $S_{6,1}$, and (d) type $S_{6,2}$.

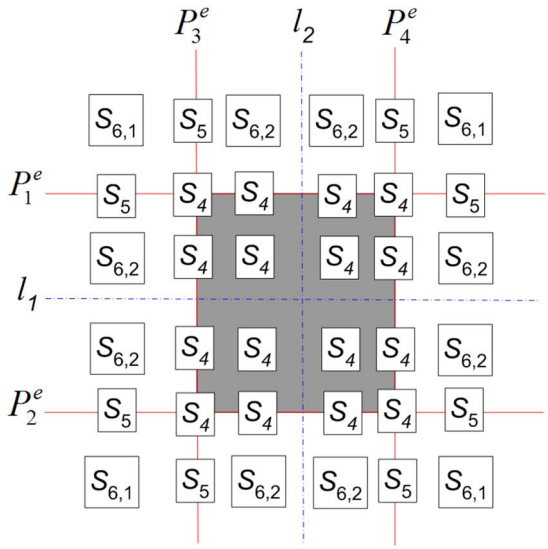


Fig. 14. Shape type of the GDM outline.

TABLE II
CDM

n_v	ST	D^s
6	$S_{6,1}$	'$BaCbAc$'
		'$aBcAbC$'
	$S_{6,2}$	'$aBCbAc$'
		'$aBcbAC$'
		'$ABaCbc$'
		'$BcAbaC$'
5	S_5	'$\underline{aB}CbAc$'
		'$aBc\underline{bA}C$'
		'$BaCbc\underline{A}$'
		'$BcAb\underline{aC}$'
4	S_4	'$\underline{aB}Cbc\underline{A}$'
		'$BCbc\underline{Aa}$'
		'$Bcb\underline{AaC}$'
		'$Bcb\underline{A}\ \underline{aC}$'
		'$\underline{AaB}Cbc$'
		'$Bcb\underline{AaC}$'
		'$BCbc$'
		'$BcbC$'

For the two cases of $n_v = 6$ in Fig. 7(e) and (f), due to the fact that their missing lines will be recovered differently when there is occlusion, they are modeled as two different types: 1) $S_{6,1}$ and 2) $S_{6,2}$ [Fig. 13(c) and (d)]. The shape type of the projected GDM is depicted in Fig. 14. The summary of the description D^s, shape type ST, and n_v can be found in Table II.

D. RM

The RM formulates the maximum number of lines of the GDM contour that are allowed to be completely or partially occluded in the image such that the original shape of the

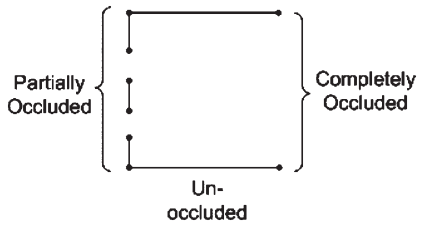


Fig. 15. States of the lines in partial occlusion.

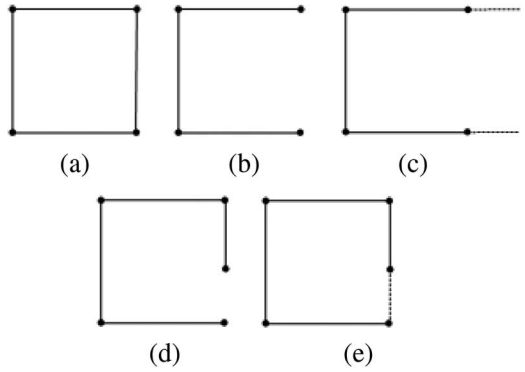


Fig. 16. S_4's resolvability. (a) Original GDM in S_4. (b) $n_{\rm occ} = 1$ in S_4. (c) (b) is irresolvable. (d) $n_p = 1$ and $n_{\rm occ} = 0$ in S_4. (e) (d) is resolvable.

GDM is still recoverable based on the information given by the unoccluded and partially occluded lines. In an unoccluded case, the projected GDM onto the image consists of a full description of lines. When there is partial occlusion, lines on the GDM can take on one of the following three states: 1) unoccluded; 2) partially occluded; and 3) completely occluded (Fig. 15).

These three states could happen on each line of the four shape types that were introduced in the previous section (i.e., S_4, S_5, $S_{6,1}$, and $S_{6,2}$). In order to derive a set of rules to model the resolvability of GDM, we have generated all the possible combinations of the three states (i.e., unoccluded, partially occluded, and completely occluded) on each of the four shape types (S_4, S_5, $S_{6,1}$, and $S_{6,2}$) and attempted to complete the missing lines for the generated samples based on the existing lines. Based on the observation of the huge number of generated cases, four conclusions are drawn.

1) For S_4 [Fig. 16(a)], if there is *one or more* completely occluded lines (i.e., $n_{\rm occ} \geq 1$, where $n_{\rm occ}$ is the number of completely occluded lines), then the GDM is irresolvable. An example is illustrated in Fig. 16(b), in which there is one completely occluded line in S_4 (i.e., $n_{\rm occ} = 1$); therefore, there is no way to recover the original GDM with the information given by the nonoccluded lines, as indicated by the dotted lines in Fig. 16(c). On the other hand, the number of partially occluded lines n_p does not affect its resolvability. This is illustrated in Fig. 16(d), in which there is only one partially occluded line in S_4 (i.e., $n_p = 1$ and $n_{\rm occ} = 0$). Therefore, it is resolvable, as indicated by the dotted lines in Fig. 16(e).
2) Similarly, in S_5 [Fig. 17(a)], if there is *one or more* completely occluded lines (i.e., $n_{\rm occ} \geq 1$), then the GDM is irresolvable. An example is shown in Fig. 17(b), in which there is one completely occluded line $(n_{\rm occ} = 1)$, indicating that the original GDM is not resolvable, even if the information of one of the vertices is exploited, as indicated by the horizontal dotted line [Fig. 17(c)]. On the other hand, similar to S_4, the number of partially occluded lines n_p does not affect its resolvability. This is illustrated in Fig. 17(d), in which the GDM contains three partially occluded lines but no completely occluded lines $(n_p = 3$ and $n_{\rm occ} = 0)$; therefore, the original shape is recoverable, as depicted in Fig. 17(e).

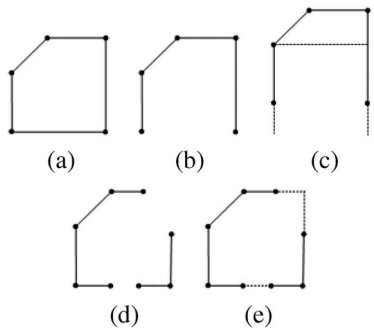


Fig. 17. S_5's resolvability. (a) Original GDM in S_5. (b) $n_{\rm occ} = 1$ in S_5. (c) (b) is irresolvable. (d) $n_p = 3$ and $n_{\rm occ} = 0$ in S_5. (e) (d) is resolvable.

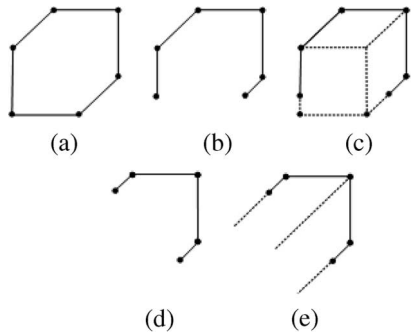


Fig. 18. $S_{6,1}$'s resolvability. (a) Original GDM in $S_{6,1}$. (b) $n_{\rm occ} = 1$ in $S_{6,1}$. (c) (b) is resolvable. (d) $n_{\rm occ} = 2$ in $S_{6,1}$. (e) (d) is irresolvable.

3) In $S_{6,1}$ [Fig. 18(a)], this case is slightly different in that if there are *more than one* completely occluded line (i.e., $n_{\rm occ} > 1$), then the GDM is irresolvable. As illustrated in Fig. 18(b), the GDM contains one completely occluded line $(n_{\rm occ} = 1)$, which is recoverable with the aid of extending the partially occluded lines and projecting the lines to form a complete GDM, as depicted by the dotted lines in Fig. 18(c). However, if $n_{\rm occ} \geq 2$ [Fig. 18(d)], the GDM becomes irresolvable, as there is no way to determine the extent of the GDM [Fig. 18(e)].
4) The case of $S_{6,2}$ is, by far, the most complicated. If, in description D^s, either "b" or "B" is between "a" and "A" [Fig. 19(a)], then, if both "b" and "B" are not completely occluded and $n_{\rm occ} \leq 1$, the GDM is resolvable. An example is depicted in Fig. 19(b), in which "b" is not occluded, "B" is only partially occluded (i.e., both of them are not completely occluded), and $n_{\rm occ} = 1$; thus, the GDM is resolvable [Fig. 19(c)]. On the other hand, if either "b" or "B" is completely occluded, or if both "b" and "B" are not completely occluded but $n_{\rm occ} > 1$, then the GDM is irresolvable. This is illustrated in Fig. 19(d) and (e), in which "B" is completely occluded;

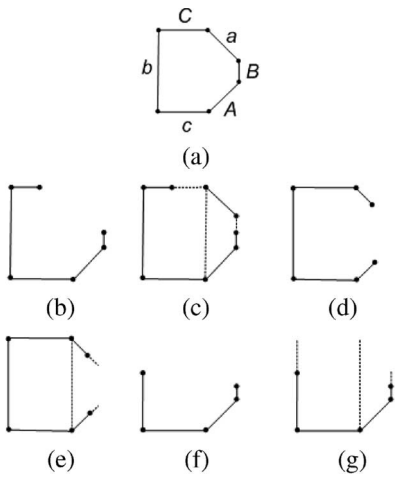


Fig. 19. $S_{6,2}$'s resolvability. (a) Original GDM in $S_{6,2}$ where "B" is between "a" and "A." (b) "b" and "B" exist and $n_{\rm occ} = 1$. (c) (b) is resolvable. (d) "B" is completely occluded. (e) (d) is irresolvable. (f) "b" and "B" exist and $n_{\rm occ} = 2$. (g) (f) is irresolvable.

TABLE III
RM

ST	R = 1 (Resolvable) if	R = 0 (Irresolvable) if
S_4 or S_5	$n_{occ} = 0$, n_p = (any value)	otherwise
$S_{6,1}$	$n_{occ} \le 1$, n_p = (any value)	otherwise
$S_{6,2}$	(i) – (iii) are all satisfied	(i) is satisfied, but (ii) is violated or (i) and (ii) are satisfied, but (iii) is violated

(i) 'b' or 'B' is between 'a' and 'A'
(ii) both 'b' and 'B' exists
(iii) $n_{occ} \le 1$
(The same condition holds when 'b' and 'B' are replaced by 'c' and 'C' respectively.)

therefore, it is irresolvable. Similarly, in Fig. 19(f), although both "b" and "B" exists, as $n_{\rm occ} = 2$, it is irresolvable [Fig. 19(g)]. (The same condition holds when "b" and "B" are replaced by "c" and "C," respectively).

The conditions that were discussed are summarized in Table III.

E. Model Application

In this section, the steps to apply the proposed models (i.e., VSM, CDM, and RM) are introduced [21]. The proposed steps are illustrated through a real-world example of vehicle occlusion, which contains a cluster of six occluded vehicles (i.e., $N = 6$) (Fig. 20).

1) Vehicle Segmentation: The texture-based vehicle segmentation approach that was described in [29] is employed to extract the moving vehicles and contains three steps. First, three likelihood maps, namely 1) T-map; 2) L-map; and 3) C-map, are computed according to the differences in texture, luminance, and chrominance between input frame f_i and background reference frame f_b, respectively. Then, a logical OR operation is performed on the likelihood maps to produce the OR-map. Finally, morphological operations are performed on the OR-map to form the foreground mask, which is used to extract the vehicle. This method has an inherent advantage that

Fig. 20. Example of vehicle occlusion ($N = 6$).

cast shadow regions are automatically removed, as they have the same textural properties as the background.

To construct the T-map, each pixel with its neighborhood from an input frame f_i is transformed into p_i, as given by

$$p_{(x,y)}(m,n) = f(x+m-M, y+n-N) \tag{1}$$

where $0 \le m \le 2M$, and $0 \le n \le 2N$, and the same neighborhood from a background reference frame f_b is transformed similarly into p_b. T-map is constructed according to texture difference d_T, i.e.,

$$d_T(x,y) = \frac{1}{(2M+1)(2N+1)} \times \sum_{u=0}^{2M}\sum_{v=0}^{2N}\left[R_{(x,y),i}(u,v) - R_{(x,y),b}(u,v)\right]^2 \tag{2}$$

where $M \le x \le X - M - 1$, and $N \le y \le Y - N - 1$, which is the mean square difference of two autocorrelation functions R of p_i and p_b. T-map is finally computed by comparing texture threshold τ_T with texture difference map d_T, i.e.,

$$T\text{-map}(x,y) = \begin{cases} 1, & \text{if } d_T(x,y) > \tau_T \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

where $M \le x \le X - M - 1$, and $N \le y \le Y - N - 1$.

To construct the L-map and C-map, the color model YCbCr is used to separate the luminance and chrominance components of the images. A luminance difference d_Y between input frame f_i and background reference frame f_b is computed according to the following equation:

$$d_Y(x,y) = \begin{cases} Y_i(x,y) - Y_b(x,y), & \text{if } Y_i(x,y) - Y_b(x,y) > 0 \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

where $M \le x \le X - M - 1$, and $N \le y \le Y - N - 1$. The chrominance difference d_C between input frame f_i and

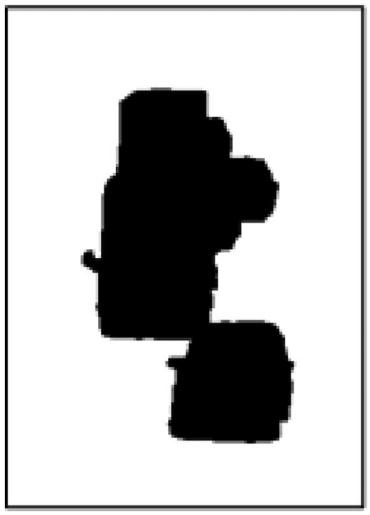


Fig. 21. Composite vehicle mask.

background reference frame f_b is computed according to the following equation:

$$d_C(x,y) = [Cb_i(x,y) - Cb_b(x,y)]^2 + [Cr_i(x,y) - Cr_b(x,y)]^2 \quad (5)$$

where $M \le x \le X - M - 1$, and $N \le y \le Y - N - 1$. Both the L-map and C-map are then calculated by comparing luminance threshold τ_Y and chrominance threshold τ_C with luminance difference map d_Y and chrominance difference map d_C, respectively, i.e.,

$$L\text{-map}(x,y) = \begin{cases} 1, & \text{if } d_Y(x,y) > \tau_Y \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

$$C\text{-map}(x,y) = \begin{cases} 1, & \text{if } d_C(x,y) > \tau_C \\ 0, & \text{otherwise} \end{cases} \quad (7)$$

where $M \le x \le X - M - 1$, and $N \le y \le Y - N - 1$.

The OR-map is then computed by a basic logical OR operation of the T-map, L-map, and C-map, i.e.,

$$\text{OR-map}(x,y) = T\text{-map}(x,y) + L\text{-map}(x,y) + C\text{-map}(x,y) \quad (8)$$

where $M \le x \le X - M - 1$, and $N \le y \le Y - N - 1$. Finally, the OR-map is operated on by a morphological opening to remove undesired features such as the boundary between the shadowed and non-shadowed regions of the road and to smooth the contours. Afterward, morphological closing is employed to fuse narrow breaks and long thin gulfs, and eliminate small holes in order to form the foreground mask. The vehicle is then extracted based on the shape of the foreground mask. This method proves to be very effective in extracting vehicle details without shadow and other visual artifacts. The composite vehicle mask of the clustered vehicles extracted from the image by this method is shown in Fig. 21.

2) Contour Approximation: To approximate the contour of the extracted vehicle mask (Fig. 21), the tangential slope curvature detection method proposed by [11] is employed to extract the curvature points due to its simplicity and ability to detect feature points under noise.

a) Computation of signature curve: The tangential slope of every data sample on the contour of the segmented object is first computed [Fig. 22(a) and (b)]. For simplicity, the tangential slope of data sample i is defined as the vector that joins samples $i-1$ and $i+1$. The signature curve of the contour, which is a set of tangential slopes, can be written mathematically as

$$s_i(x_i, y_i) = \frac{y_{i+1} - y_{i-1}}{x_{i+1} - x_{i-1}} \quad (9)$$

for $i = 2, \ldots, N_s - 1$, where N_s is the number of sample points on the contour, and x_i and y_i are the x- and y-components of the ith sample point, respectively. The merit of this method is that it is straightforward and retains points of sharp and gradual curvatures. However, this also implies that the signature curve also contains a considerable amount of local variations or noise [Fig. 22(c)] that required suppression in the next stage.

b) Signature curve filtering: These local variations or noise are due to quantization in the digitizing process of the contour, and they must be filtered before the curvature points can be detected from the signature curve. From a detailed study of the signature curve, it is found that most of these local variations have a duration of not more than two samples in length. As such, the filter can simply eliminate any spikes with a duration of *less than or equal to* two samples in length. In another words, the filter changes the signal $[\ldots, p, q, p, \ldots]$ to $[\ldots, p, p, p, \ldots]$, and $[\ldots, p, q, q, p, \ldots]$ to $[\ldots, p, p, p, p, \ldots]$, respectively, where p and q are two different tangential slope values. Mathematically, it can be written as

$$f_i = s_{i-1} \text{ if } (s_i \ne s_{i-1} \text{ and } s_{i+1} = s_{i-1}) \quad (10)$$

$$f_i = s_{i-1} \text{ and } f_{i+1} = s_{i-1} \text{ if } (s_i \ne s_{i-1}, s_{i+1} = s_i, \text{ and } s_{i+2} = s_{i-1}) \quad (11)$$

where s_i and f_i are the tangential slope value of the signature curve and filtered signature curve, respectively. The resulting filtered signature curve is depicted in Fig. 22(d). Comparing this with Fig. 22(b), all the local variations have been removed.

c) Curvature detection: To detect the curvature points from the filtered signature curve, the following rules are used. First, the start and end points of each disjointed segment on the filtered signature curve are considered as curvature points since they represent a substantial change in tangential slopes. Second, curvature points are also detected at the transition point where the filtered signature curve changes state. This can be written mathematically as follows:

$$\text{if } (f_{i-1} = \pm\infty, f_i \ne \pm\infty, \text{ and } f_{i+1} \ne \pm\infty) \text{ then } \mathbf{c}_i = [x_i \quad y_i]^T \text{and } \mathbf{c}_{i+1} = [x_{i+1} \quad y_{i+1}]^T \text{ if } f_i \ne f_{i+1} \quad (12)$$

$$\text{if } (f_{i-1} \ne \pm\infty, f_i \ne \pm\infty, \text{ and } f_{i+1} = \pm\infty) \text{ then } \mathbf{c}_i = [x_{i-1} \quad y_{i-1}]^T \text{ and } \mathbf{c}_{i+1} = [x_i \quad y_i]^T \text{ if } f_{i-1} \ne f_i \quad (13)$$

$$\text{if } (f_{i-1} = f_i \ne \pm\infty \text{ and } f_{i+1} = f_{i+2} \ne \pm\infty) \text{ then } \mathbf{c}_i = [x_i \quad y_i]^T \text{ and } \mathbf{c}_{i+1} = [x_{i+1} \quad y_{i+1}]^T \text{ if } f_i \ne f_{i+1} \quad (14)$$

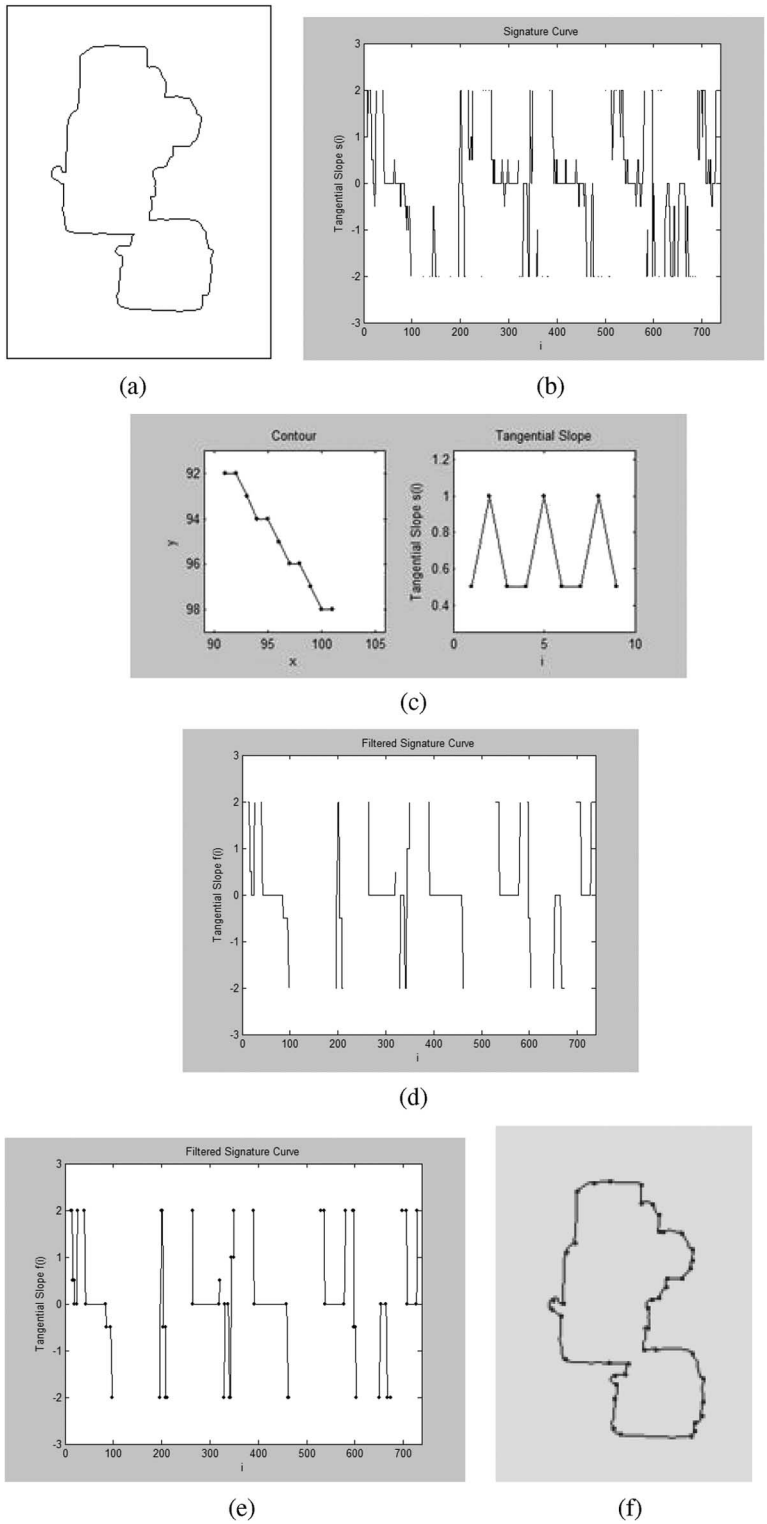


Fig. 22. Tangential slope curvature points extraction of the vehicle cluster. (a) Composite contour, (b) signature curve of (a), (c) local variations on the signature curve, (d) filtered signature curve, (e) detected curvature points on curve, and (f) curvature points detected.

where $\mathbf{c}_i$ is the detected curvature point at $[x_i \ y_i]^T$ in the 2-D image coordinates. The detected curvature points are shown in Fig. 22(e). For clustered curvature points, we simply select one point with the highest curvature out of a cluster of detected points. The set of final curvature points is depicted in Fig. 22(f), and we can observe from this diagram that the proposed algorithm has accurately detected the curvature points. The set of detected curvature points can be written as $\mathbf{c} = [\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_K]^T$, where K is the number of detected curvature points. This results in the approximated contour for the subsequent process.

3) Count Reasoning: With the approximated contour in hand, it is then further reasoned in order to find out how many vehicles are involved in the contour. To do this, the

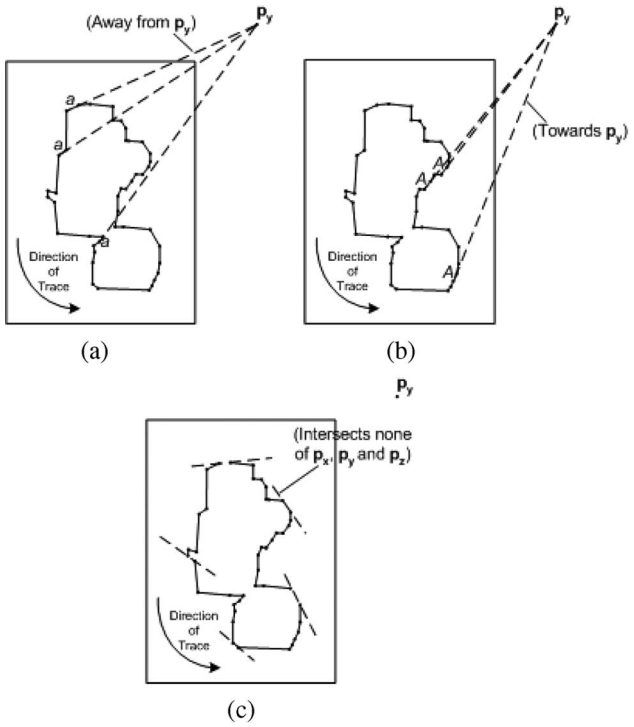


Fig. 23. Contour labeling by direction primitives. (a) Direction primitive "a," (b) direction primitive "A," and (c) lines without direction primitives.

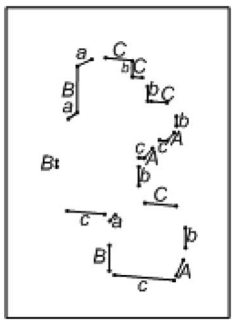


Fig. 24. Contour with description D^s.

lines of the approximated contour are first described by direction primitives (i.e., "a," "b," "c," "A," "B," and "C") (Section IV-C), depending on to which of the vanishing points (i.e., $\mathbf{p_x}$, $\mathbf{p_y}$, or $\mathbf{p_z}$) the line relates. The line of the approximated contour is extended to see if the extend line intersects $\mathbf{p_x}$, $\mathbf{p_y}$, or $\mathbf{p_z}$. If the extend line intersects $\mathbf{p_y}$, and when the contour is traced in a counterclockwise manner, the direction of the line is *away from* $\mathbf{p_y}$, then it is labeled as "a" [Fig. 23(a)]. On the other hand, if the direction of the line goes *toward* $\mathbf{p_y}$ when the contour is traced in a counterclockwise manner, then it is labeled as "A" [Fig. 23(b)]. The labeling of "b," "B," "c," and "C" are done in the same manner. If the extend line does not intersect any of the vanishing points, then it is simply ignored and deleted from the contour [Fig. 23(c)]. If the extend line intersects more than one vanishing point, then the underlined convention (e.g., $\underline{aC}$ and $\underline{AaC}$) is used, as described in Section IV-C. By tracing the approximated contour in this way, the description $D^s =$ "$aBaBcaBcAbCbcAcAbCbCbC$" is obtained (Fig. 24).

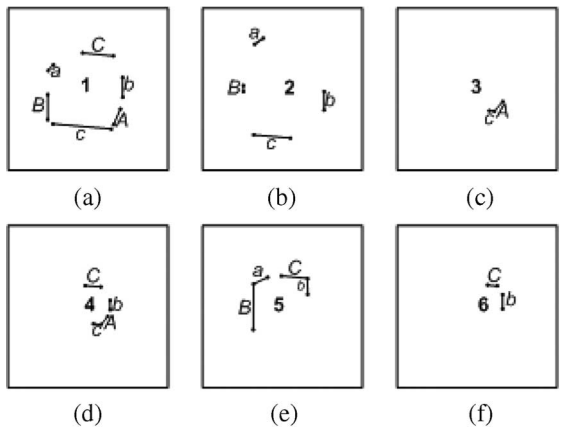


Fig. 25. Segmented subsegments. (a) D_1^s, (b) D_2^s, (c) D_3^s, (d) D_4^s, (e) D_5^s, and (f) D_6^s.

In order to reason how many vehicles are involved within the approximated contour, description D^s is subdivided into subsegments D_i^s, with reference to the CDM (i.e., Table II of Section IV-C), which represents a partially completed GDM. D_i^s is segmented from D^s if it matches one of the cases of Table II. For description D^s (Fig. 24), six subsegments $(D_1^s, \ldots, D_6^s)$ are being segmented, which are depicted here.

1) $D_1^s =$ "$aBcAbC$," which corresponds to "$aBcAbC$" $(ST = S_{6,1})$.
2) $D_2^s =$ "$aBc \ldots b \ldots$," which corresponds to "$aBcAbC$" $(ST = S_{6,1})$.
3) $D_3^s =$ "$\ldots cA \ldots$," which corresponds to "$aBcAbC$" $(ST = S_{6,1})$.
4) $D_4^s =$ "$\ldots cAbC \ldots$," which corresponds to "$aBcAbC$" $(ST = S_{6,1})$.
5) $D_5^s =$ "$aB \ldots bC$," which corresponds to "$aBcAbC$" $(ST = S_{6,1})$.
6) $D_6^s =$ "$\ldots bC \ldots$," which corresponds to "$aBcAbC$" $(ST = S_{6,1})$.

In another words, six vehicles are being counted from the cluster, which is a correct count (Fig. 25).

4) Resolvability Evaluation: After the description D^s of the approximated contour is computed and segmented into D_i^s, with reference to the CDM, the next step is to assign resolvability index R to each subsegment according to the criteria in the RM (Table III), which indicates the likelihood of success to recover the original GDM of the vehicle. In comparison with the projection of the GDM in the CDM (Table II), the number of completely occluded lines n_{occ} (indicated as "complete") and partially occluded lines n_p (indicated as "partial") is given here.

1) In $D_1^s =$ "$aBcAbC$," the status of the lines are "a (partial) B (partial) c (partial) A (partial) b (partial) C (partial)." Therefore, $n_{\text{occ}} = 0$, and $n_p = 6$.
2) In $D_2^s =$ "$aBc \ldots b \ldots$," the status of the lines are "a (partial) B (partial) c (partial) A (complete) b (partial) C (complete)." Therefore, $n_{\text{occ}} = 2$, and $n_p = 4$.
3) In $D_3^s =$ "$\ldots cA \ldots$," the status of the lines are "a (complete) B (complete) c (partial) A (partial) b (complete) C (complete)." Therefore, $n_{\text{occ}} = 4$, and $n_p = 2$.

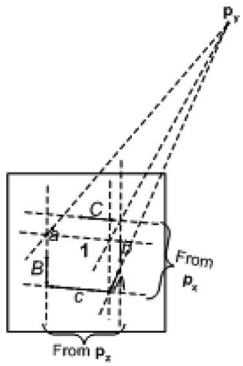


Fig. 26. Resolving the GDM of vehicle 1.

4) In $D_4^s =$ "$\ldots cAbC \ldots,$" the status of the lines are "a (complete) B (complete) c (partial) A (partial) b (partial) C (partial)." Therefore, $n_{\text{occ}} = 2$, and $n_p = 4$.
5) In $D_5^s =$ "$aB \ldots bC,$" the status of the lines are "a (partial) B (partial) c (complete) A (complete) b (partial) C (partial)." Therefore, $n_{\text{occ}} = 2$, and $n_p = 4$.
6) In $D_6^s =$ "$\ldots bC \ldots,$" the status of the lines are "a (complete) B (complete) c (complete) A (complete) b (partial) C (partial)." Therefore, $n_{\text{occ}} = 4$, and $n_p = 2$.

Based on the status of the lines of each subsegment, resolvability index R is computed based on the rules in the RM (Table III), which are depicted here.

1) For D_1^s, since $ST = S_{6,1}$, $n_{\text{occ}} = 0$, and $n_p = 6$, $R = 1$, because $n_{\text{occ}} \leq 1$ in $S_{6,1}$.
2) For D_2^s, since $ST = S_{6,1}$, $n_{\text{occ}} = 2$, and $n_p = 4$, $R = 0$, because $n_{\text{occ}} > 1$ in $S_{6,1}$.
3) For D_3^s, since $ST = S_{6,1}$, $n_{\text{occ}} = 4$, and $n_p = 2$, $R = 0$, because $n_{\text{occ}} > 1$ in $S_{6,1}$.
4) For D_4^s, since $ST = S_{6,1}$, $n_{\text{occ}} = 2$, and $n_p = 4$, $R = 0$, because $n_{\text{occ}} > 1$ in $S_{6,1}$.
5) For D_5^s, since $ST = S_{6,1}$, $n_{\text{occ}} = 2$, and $n_p = 4$, $R = 0$, because $n_{\text{occ}} > 1$ in $S_{6,1}$.
6) For D_6^s, since $ST = S_{6,1}$, $n_{\text{occ}} = 4$, and $n_p = 2$, $R = 0$, because $n_{\text{occ}} > 1$ in $S_{6,1}$.

5) Resolving GDM: After the computation of the resolvability index of each vehicle, the final step is to resolve the GDM of each vehicle [11]. To do this, the missing lines of the GDM are recovered geometrically according to the existing nonoccluded lines of the GDM as well as vanishing points $\mathbf{p_x}$, $\mathbf{p_y}$, and $\mathbf{p_z}$, as illustrated in Fig. 26, which shows how the GDM of vehicle 1 is reconstructed. The reconstructed GDM is "$aBcAbC$." The results of all the reconstructed GDMs are shown in Fig. 27. The segmentation of each of the vehicles is depicted in Fig. 28. It can be noted from Fig. 28(a) that the sedan (i.e., vehicle 1) is perfectly segmented from the image since it is declared as resolvable. On the other hand, for the rest of the vehicles [i.e., the truck at the front (vehicle 2), the taxi at the front (vehicle 3), the taxi in the middle (vehicle 4), the truck at the back (vehicle 5), and the taxi at the back (vehicle 6)], due to the fact that they are declared as irresolvable by the RM, therefore, their GDMs are not fully recoverable [Fig. 28(b)–(f)]. However, they were successfully counted.

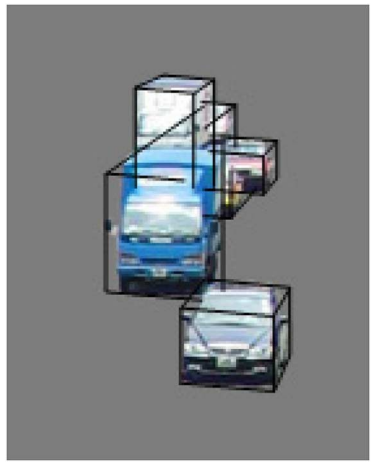


Fig. 27. Resolved GDMs of all vehicles.

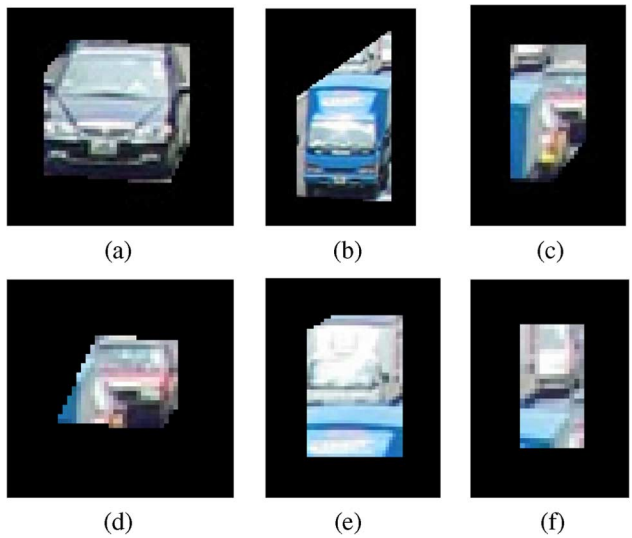


Fig. 28. Segmented vehicles based on the resolved GDM. (a) Sedan, (b) truck, (c) taxi, (d) taxi, (e) truck, and (f) taxi.

6) Vehicle Dimension Estimation: From the vertices of the resolved GDM $\mathbf{v} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_6]^T$, vehicle dimensions are determined by transforming them from 2-D image coordinates to 3-D world coordinates using the camera parameters that are determined through camera calibration [30]. The model vertices after the transformation are denoted by the capital letter $\mathbf{V}$, i.e., $\mathbf{V} = [\mathbf{V}_1, \mathbf{V}_2, \ldots, \mathbf{V}_6]^T$. The dimensions of the resolved model are then estimated as follows:

$$W = |\mathbf{V}_4 - \mathbf{V}_3| \tag{15}$$

$$L = |\mathbf{V}_5 - \mathbf{V}_4| \tag{16}$$

$$H = |\mathbf{V}_6 - \mathbf{V}_5| \tag{17}$$

where W, L, and H stand for the width, length, and height of the vehicle, respectively. The dimensions of the six vehicles of the real-world example are extracted as depicted in Table IV. It should be noted that some of the dimensions of the vehicles could not be estimated since they are declared as irresolvable. The extracted dimensions are compared with the dimension, as specified by the manufacturer (Table V), and

TABLE IV
ESTIMATED DIMENSIONS OF THE REAL-WORLD EXAMPLE

	Width (W) (mm)	Length (L) (mm)	Height (H) (mm)
Sedan (Vehicle 1)	1733	4292	1435
Truck at the front (Vehicle 2)	2327	–	2418
Taxi at the front (Vehicle 3)	–	–	–
Taxi in the middle (Vehicle 4)	–	4758	1470
Truck at the back (Vehicle 5)	2199	5949	–
Taxi at the back (Vehicle 6)	–	–	–

TABLE V
DIMENSIONS SPECIFIED BY THE MANUFACTURER

	Width (W) (mm)	Length (L) (mm)	Height (H) (mm)
Sedan (Honda Civic)	1725	4418	1413
Truck (Toyota Hino)	2240	6115	2505
Taxi (Toyota Crown)	1765	4820	1465

TABLE VI
PERCENTAGE ERROR OF THE REAL-WORLD EXAMPLE

	Width (W)	Length (L)	Height (H)
Sedan (Vehicle 1)	0.46%	2.85%	1.56%
Truck at the front (Vehicle 2)	3.88%	–	3.47%
Taxi at the front (Vehicle 3)	–	–	–
Taxi in the middle (Vehicle 4)	–	1.29%	0.34%
Truck at the back (Vehicle 5)	1.83%	2.71%	–
Taxi at the back (Vehicle 6)	–	–	–

the percentage error of the estimation is computed by dividing the absolute dimension error by the actual dimension, as listed in Table VI.

V. RESULT AND DISCUSSION

The proposed method was evaluated on 267 sets of real-world monocular traffic images that were taken on a busy urban road that contains 3074 vehicles with multiple-vehicle occlusions that can be seen in the image. The texture-based segmentation approach described in [29] was employed to extract the composite contour of the occluded vehicles and eliminate the vehicle cast shadow. The scoreboard algorithm in [28] was employed to estimate the background image for vehicle segmentation and to cope with background changing due to roadside objects. The extracted contour was first approximated by the curvature detection method that was proposed by Pang *et al.* [11]. The number of vehicles that were involved in each case was deduced with reference to the CDM. By comparing the vehicle count that was obtained by human inspection in monocular images with the one obtained by the proposed method, it was found that the proposed method has achieved a count accuracy of 100%. It should be noted that this percentage does not include those vehicles that are being totally occluded by the image, as there is no way to detect such vehicles in monocular images by human inspection nor by the proposed method. The resolvability index of each vehicle was evaluated with reference to the RM. The individual GDM of the vehicles was computed [21]. In this case, 93.82% of the vehicles were resolvable, while the rest were irresolvable, although they were included in the count. For the irresolvable vehicles, the dimensions of the vehicle could not be correctly estimated, but they could be correctly counted. The segmented GDM was further analyzed subjectively for each case to see if the proposed method performed in a satisfactory manner (Figs. 29–35). To objectively evaluate the accuracy of the proposed method, the dimensions of the vehicles that were segmented from the

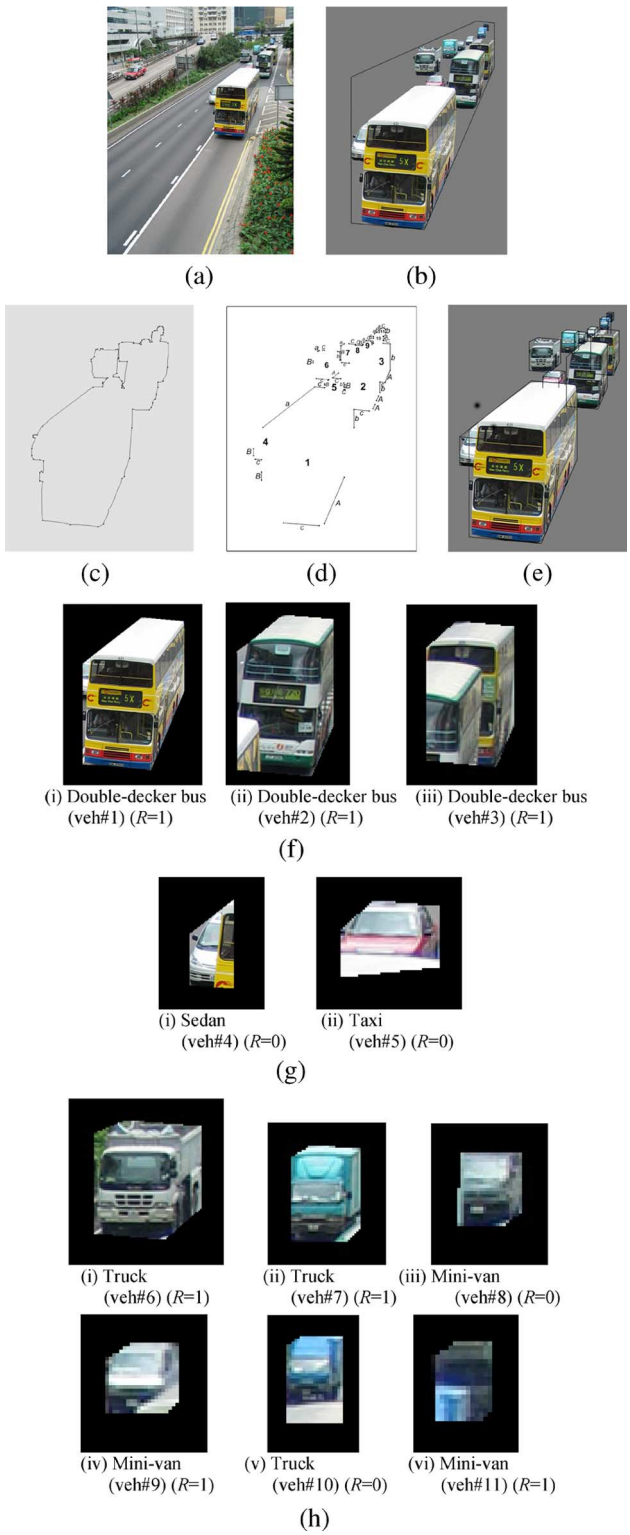


Fig. 29. Moderately occluded cluster ($N = 11$). (a) Image. (b) Composite GDM. (c) Approximated contour. (d) Computed contour. (e) Resolved GDM. (f) Right lane (from the viewer). (g) Middle lane. (h) Left lane.

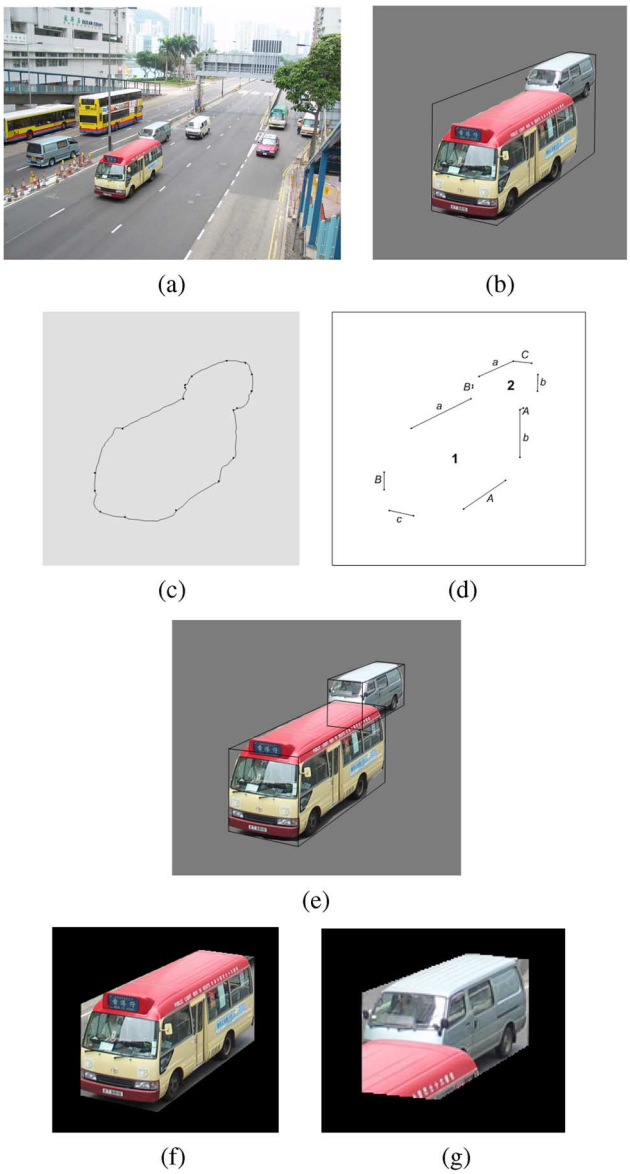


Fig. 30. Close vehicle cluster ($N = 2$). (a) Image. (b) Composite GDM. (c) Approximated contour. (d) Computed contour. (e) Resolved GDM. (f) Minibus (veh#1) ($R = 1$). (g) Minivan (veh#2) ($R = 1$).

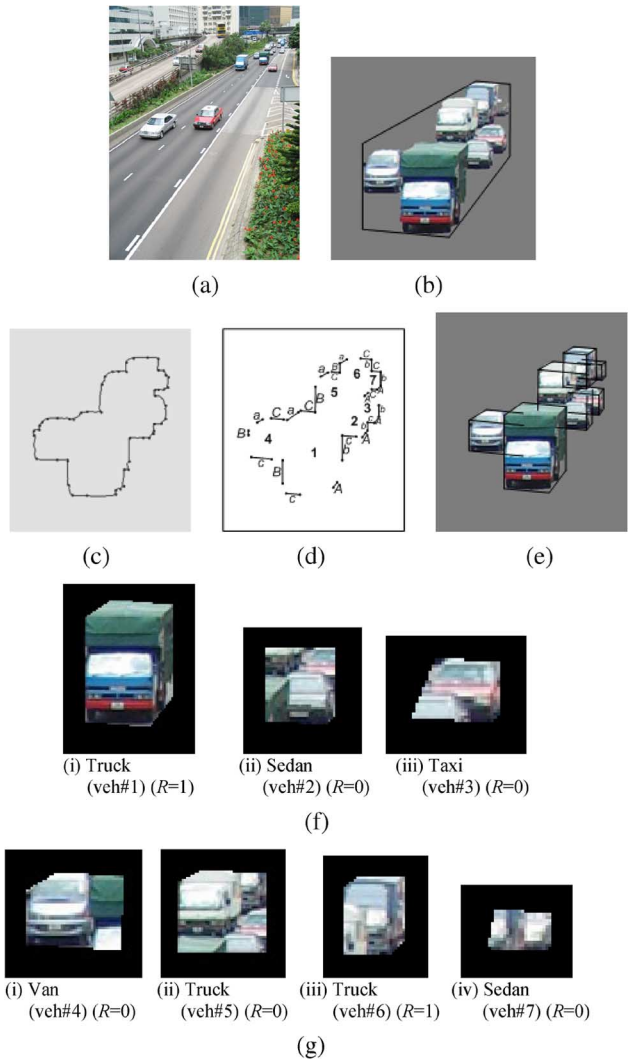


Fig. 31. Low-resolution distant vehicle cluster ($N = 7$). (a) Image. (b) Composite GDM. (c) Approximated contour. (d) Computed contour. (e) Resolved GDM. (f) Right lane. (g) Left lane.

test images were calculated from the partitioned model. The estimated values are then compared with the actual dimensions that were published by the manufacturers, and the percentage errors of the estimated dimensions are plotted in Fig. 36.

A. Moderately Occluded Cluster

The first set of images that are presented here contains a cluster of 11 vehicles traveling on three lanes that are moderately occluded [Fig. 29(a)]. The segmented vehicles and the fitted GDM for the cluster are depicted in Fig. 29(b). The approximated contour and the computed contour as well as the resolved GDMs are shown in Fig. 29(c)–(e), respectively. The segmentation of each vehicle based on the resolved GDMs are shown in Fig. 29(f)–(h). It can be seen from Fig. 29(d) that the proposed method successfully deduced the number of vehicles within the cluster as well as computed the resolvability index of each vehicle. The result is that the three buses on the right lane are resolvable, and their GDMs are accurately resolved [Fig. 29(f)]. On the other hand, the two vehicles in the middle lane are found to be irresolvable [Fig. 29(g)] due to severe occlusion. On the left lane, four out of six vehicles are resolvable [Fig. 29(h)], and their GDMs are reasonably resolved [Fig. 29(h)(i), (ii), (iv), and (vi)].

B. Close Vehicle Cluster

The second set of images that are presented here contains a cluster of two vehicles that are close to the camera [Fig. 30(a)]. The segmented vehicles and the fitted GDM for the cluster are depicted in Fig. 30(b). The approximated contour, the computed contour, and the resolved GDMs are shown in Fig. 30(c)–(e), respectively. It can be seen from Fig. 30(d) that the number of vehicles as well as the resolvability index of both vehicles is successfully deduced. The result is that both the minibus and the minivan are resolvable, and their GDMs are accurately resolved [Fig. 30(f) and (g)].

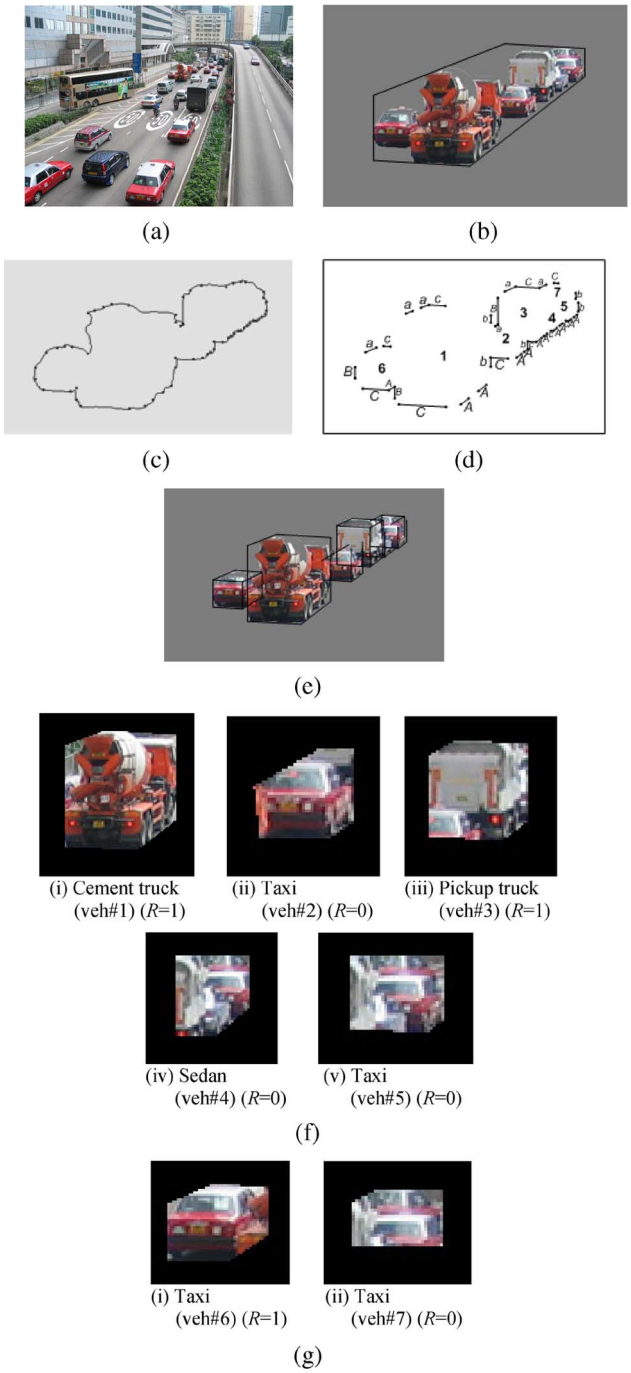


Fig. 32. Distant queue of irregular sizes ($N = 7$). (a) Image. (b) Composite GDM. (c) Approximated contour. (d) Computed contour. (e) Resolved GDM. (f) Right lane. (g) Left lane.

C. Low-Resolution Distant Vehicle Cluster

The third set of images that are presented here contains a cluster of seven vehicles that are located relatively far away in the field of view (FOV) of the image [Fig. 31(a)]. Despite the resolution of the cluster in the FOV, the proposed method has successfully computed and counted the vehicles within the cluster [Fig. 31(c) and (d)]. In terms of resolvability, only two trucks are resolvable [Fig. 31(f)(i) and (g)(iii)], while the rest are irresolvable.

D. Distant Queue of Irregular Sizes

The fourth set of images that are presented here contains a queue of seven vehicles of irregular-shaped sizes, such as a cement truck and pickup truck [Fig. 32(a)]. The GDM that is fitted onto the segmented cluster is shown in Fig. 32(b). Once again, the number of vehicles was successfully computed by the proposed method [Fig. 32(c) and (d)]. For the resolvability of the vehicles, the cement truck [Fig. 32(f)(i)], the pickup truck [Fig. 32(f)(iii)], and the taxi on the left lane [Fig. 32(g)(i)] are resolvable and accurately segmented, while the rest of the vehicles are irresolvable.

E. Large Vehicle Cluster

The fifth set of images that are presented here contains a very big cluster of 26 vehicles in four lanes [Fig. 33(a)]. Some of the vehicles are badly occluded in the image. The GDM that is fitted onto the segmented cluster is shown in Fig. 33(b). By applying the proposed method, the number of vehicles is successfully computed, including the motorcycle in Fig. 33(g)(iii). The resolved vehicles in each lane are shown in Fig. 33(f)–(i). It can be seen that the truck on the right lane [Fig. 33(f)(i)], the four taxis, the sedan, and the cement truck on the second lane from the right [Fig. 33(g)(i), (ii), (iv)–(vi), and (viii), respectively], and the taxi on the third lane from the right [Fig. 33(h)(i)] are resolvable, while the rest of the vehicles are irresolvable.

F. Rainy Situation

The sixth set of images that are presented here contains a cluster of four vehicles that are traveling on a rainy day [Fig. 34(a)]. It can be noted from Fig. 34(a) that the rain was quite heavy and the ground was wet, which causes light to be reflected on the road. Despite these effects that were brought by the rain, the proposed method successfully segmented the vehicle cluster from the background and computed the number and resolvability of the vehicles [Fig. 34(b)–(d)]. The result was that both the tour bus and the sedan on the right lane are resolvable [Fig. 34(f)], while the two sedans on the left lane are irresolvable [Fig. 34(g)].

G. Poor Illumination During Evening

The seventh set of images that are presented here contains vehicles that were taken in the evening while the illumination was very poor [Fig. 35(a)]. The images were taken without flashlight due to the fact that flashlights are impractical for traffic surveillance since it would cause unnecessary disturbance to drivers. Despite the poor illumination of the images, the vehicle cluster was successfully segmented from the image, and the number of vehicles and the resolvability of the vehicles are successfully computed [Fig. 35(b)–(d)]. The result was that the truck on the right lane and the truck on the middle lane are resolvable [Fig. 35(f) and (g)], while the taxi on the left lane is irresolvable [Fig. 35(h)].

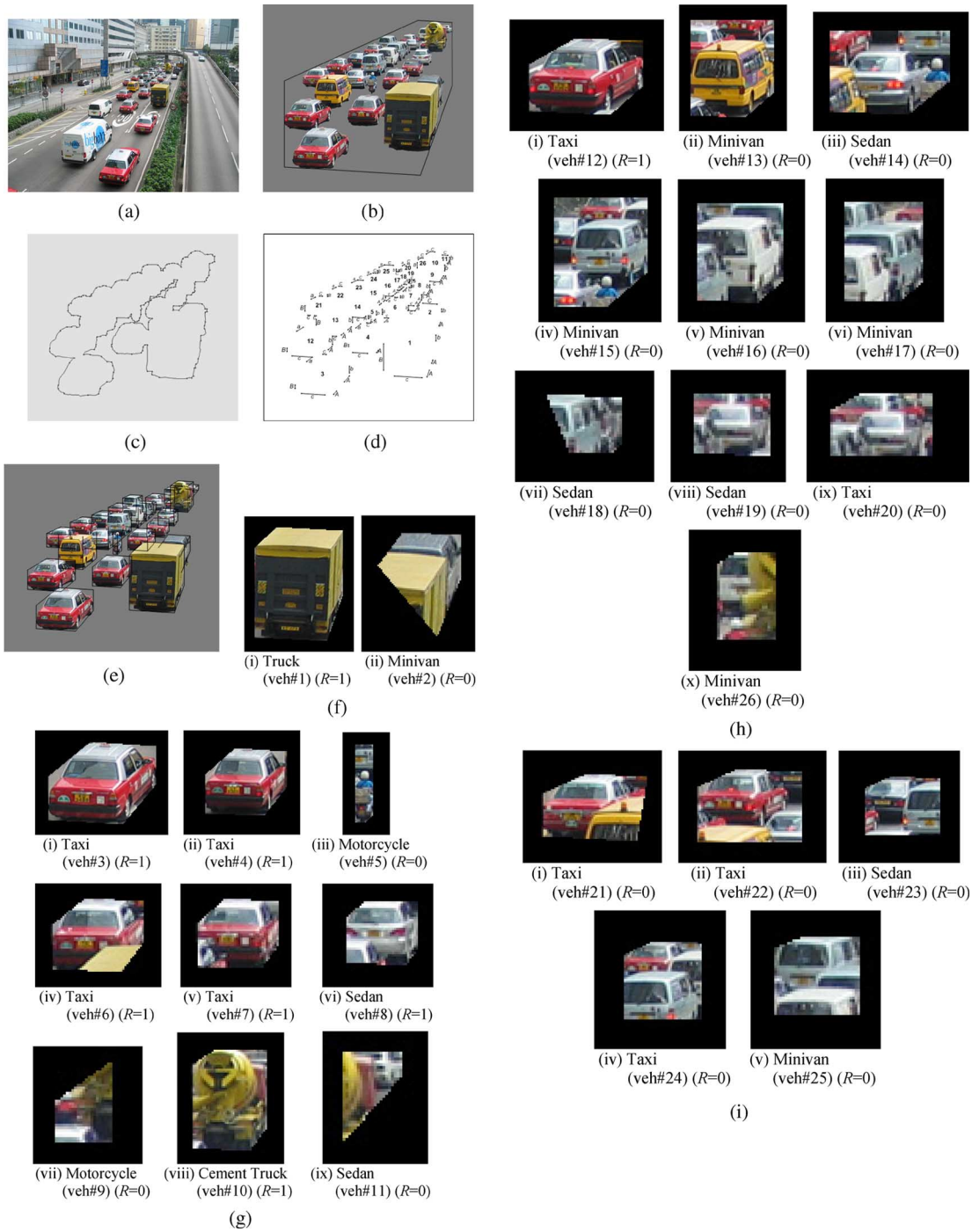


Fig. 33. Large vehicle cluster ($N = 26$). (a) Image. (b) Composite GDM. (c) Approximated contour. (d) Computed contour. (e) Resolved GDM. (f) Right lane. (g) Second lane from the right. (h) Third lane from the right. (i) Left lane (from the viewer).

H. Dimension Estimation Error

The accuracy of the proposed method is evaluated by estimating the dimensions from the resolved GDM of the vehicles in the test images. The estimated values are then compared with the actual dimensions published by the manufacturers, and the percentage error is computed by dividing the absolute dimension error by the actual dimension, as depicted in the histograms in Fig. 36. The rms error for width, length, and height estimations are found to be 48, 279, and 76 mm, respectively. In all three cases, the worst-case error is 15%, while the minimum error is about 1%. These figures indicate the potential of the proposed method.

I. Computational Time

The computational time of the proposed method is depicted in Table VII. It was implemented in MATLAB on a Pentium III 800-MHz platform, and the computational time of each step was monitored by the "tic" and "toc" functions in MATLAB. The count reasoning step is the slowest among the four steps,

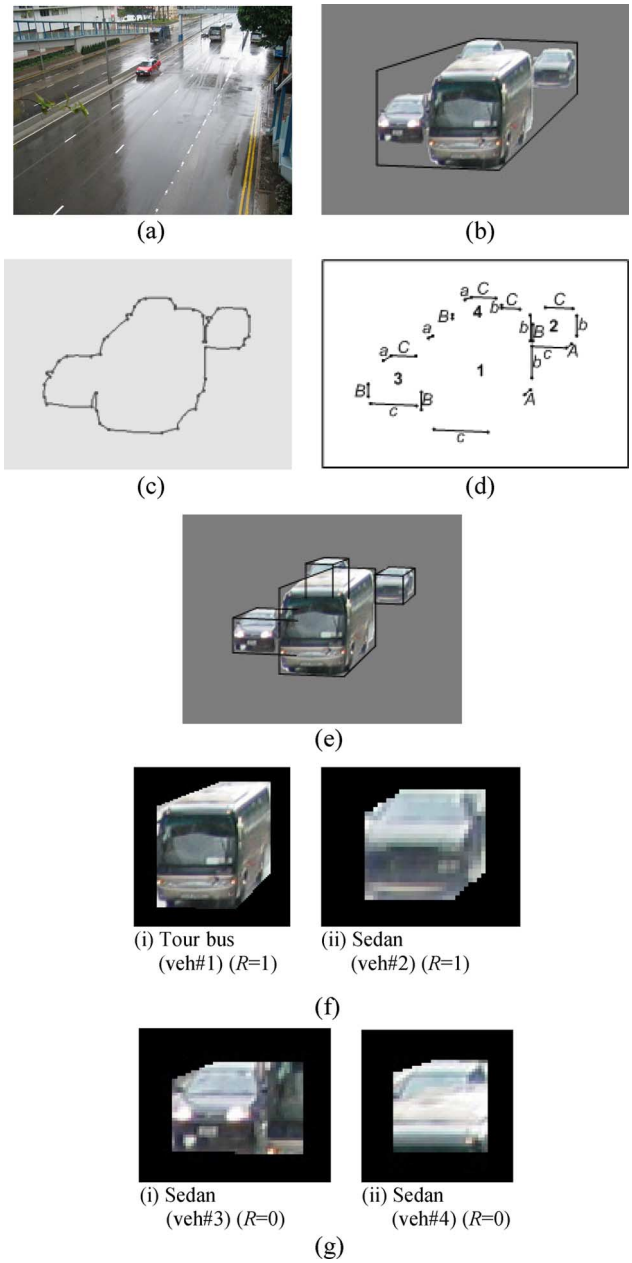


Fig. 34. Rainy example $(N = 4)$. (a) Image. (b) Composite GDM. (c) Approximated contour. (d) Computed contour. (e) Resolved GDM. (f) Right lane. (g) Left lane.

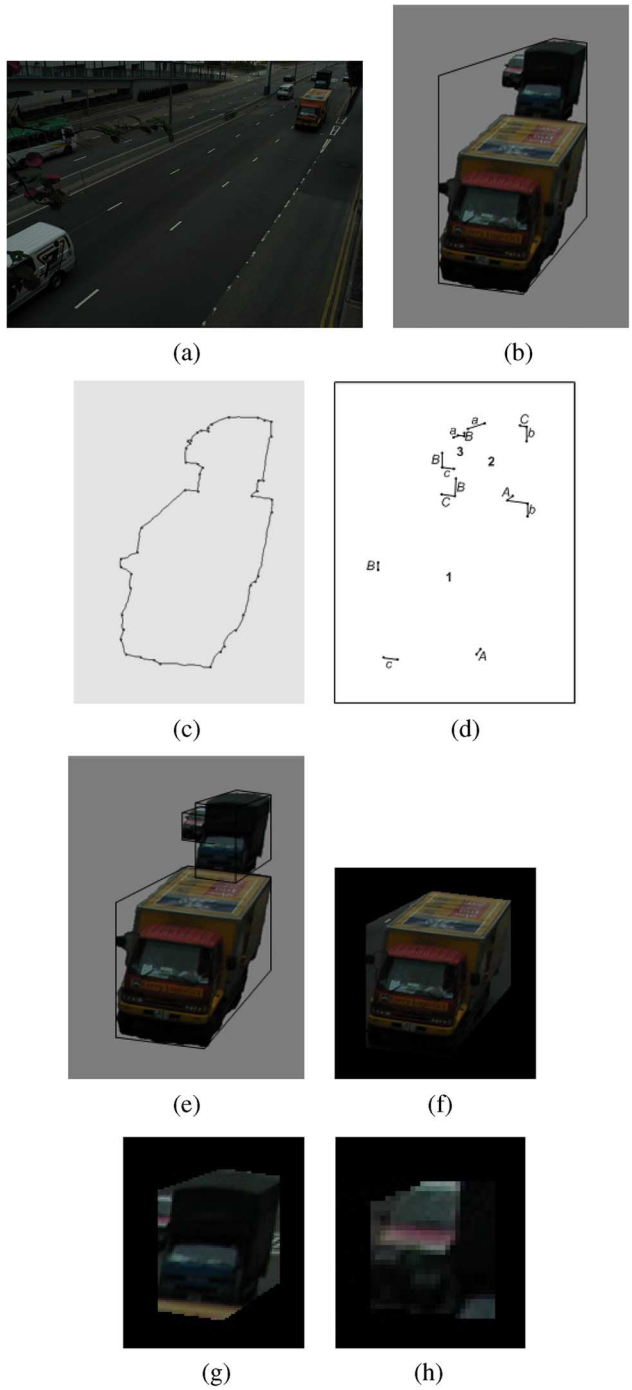


Fig. 35. Evening example $(N = 3)$. (a) Image. (b) Composite GDM. (c) Approximated contour. (d) Computed contour. (e) Resolved GDM. (f) Truck (veh#1) $(R = 1)$. (g) Truck (veh#2)$(R = 1)$. (h) Taxi (veh#3)$(R = 0)$.

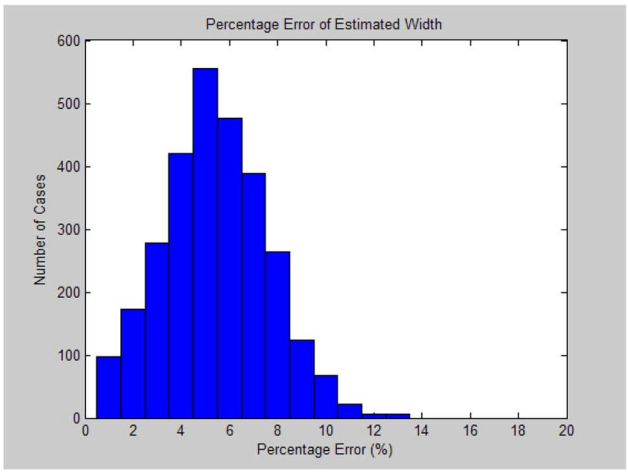
while the resolvability testing step is the fastest. The overall computational time was about 2 s/image. As such, the proposed method is computationally feasible.

VI. Suggestion for Future Development

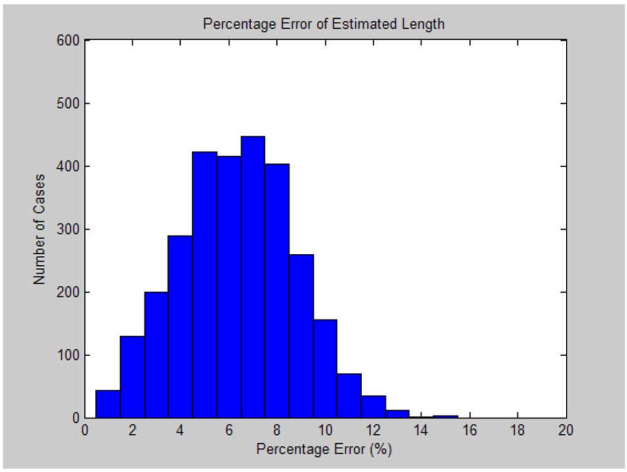
For the proposed method, an immediate question that was raised would be on how to deal with vehicles that are irresolvable. One possible solution is to exploit the symmetrical features of the vehicle such as the license plate or headlight to estimate the missing dimension of the vehicle. Another possible approach is to build a database of the relative dimension ratios of GDMs of different types of vehicles. This allows the estimation of the missing dimension of the GDM of a particular vehicle type based on the existing dimension information of the GDM of that vehicle.

VII. Conclusion

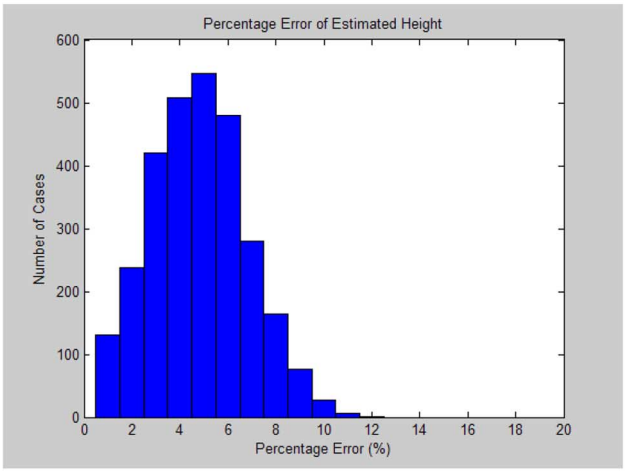
This paper proposed a novel method for accurately counting N occluded vehicles as well as determining the resolvability of the model of each vehicle. The novelty of the proposed method is that it has addressed the problem of N-vehicle occlusion as well as the severity of vehicle occlusion. The advantage of such approach is that it can reason and count the number of



(a)



(b)



(c)

Fig. 36. Percentage error of the dimension estimation of the individual vehicles. (a) Width, (b) length, and (c) height.

vehicles that are involved in a large occluded cluster under different PTZ actions of the camera, and determine whether the 3-Ds of each vehicle can be correctly estimated. As it is, accurate counting provides essential information for the estimation of important traffic parameters, while accurately resolving the vehicle from an occluded cluster is a prerequisite step for accurate tracking and classification. The proposed method has been tested on 267 sets of real-world monocular traffic images that involved multiple-vehicle occlusions and is found to be successful in counting the vehicles with a counting accuracy of 100% in comparison with human inspection. Moreover, the rms error for width, length, and height estimations are 48, 279, and 76 mm, respectively. Potentially, the proposed method can be generalized for applications such as crowd analysis, human counting, object counting, and bin picking.

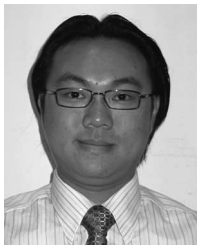
TABLE VII
COMPUTATIONAL TIME REQUIREMENT OF THE PROPOSED METHOD

	Computational time
Contour approximation	320 ms
Count reasoning	860 ms
Resolvability testing	180 ms
GDM resolving	530 ms
Overall	1890 ms

REFERENCES

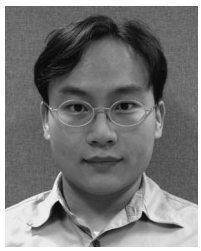
[1] W. Hu, T. Tan, L. Wang, and S. Maybank, "A survey on visual surveillance of object motion and behaviors," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 34, no. 3, pp. 334–352, Aug. 2004.
[2] K. Chen and J. C. Miles, *ITS Handbook 2000: Recommendations From the World Road Association (PIARC)*. Boston, MA: Artech House, 2000.
[3] A. H. S. Lai, "An effective methodology for visual traffic surveillance," Ph.D. dissertation, Univ. Hong Kong, Hong Kong, 2000.
[4] A. Angel, M. Hickman, P. Mirchandani, and D. Chandnani, "Methods of traffic data collection using aerial video," in *Proc. IEEE Intell. Transp. Syst. Conf.*, Sep. 2002, pp. 31–36.
[5] M. Fathy and M. Y. Siyal, "Real-time image processing approach to measure traffic queue parameters," *Proc. Inst. Electr. Eng.—Vision, Image Signal Processing*, vol. 142, no. 5, pp. 297–303, Oct. 1995.
[6] M. Fathy and M. Y. Siyal, "A window-based image processing technique for quantitative and qualitative analysis of road traffic parameters," *IEEE Trans. Veh. Technol.*, vol. 47, no. 4, pp. 1342–1349, Nov. 1998.
[7] H. S. Hsieh, K. L. Ting, and R. M. Wang, "An image processing system for signalized intersections," in *Proc. IEEE Veh. Navigat. Inf. Syst. Conf.*, Aug. 1995, pp. 28–34.
[8] J. Lou, Q. Liu, T. Tan, and W. Hu, "Semantic interpretation of object activities in a surveillance system," in *Proc. IEEE Int. Conf. Pattern Recog.*, Aug. 2002, vol. 3, pp. 777–780.
[9] S. N. Razavi and M. Fathy, "Qualitative traffic analysis using image processing and time-delayed neural network," in *Proc. IEEE Intell. Transp. Syst. Conf.*, Sep. 2002, pp. 55–60.
[10] M. Zanin, S. Messelodi, and C. M. Modena, "An efficient vehicle queue detection system based on image processing," in *Proc. IEEE Int. Conf. Image Anal. Process.*, Sep. 2003, pp. 232–237.
[11] C. C. C. Pang, W. W. L. Lam, and N. H. C. Yung, "A novel method for resolving vehicle occlusion in a monocular traffic-image sequence," *IEEE Trans. Intell. Transp. Syst.*, vol. 5, no. 3, pp. 129–141, Sep. 2004.
[12] N. H. C. Yung and A. H. S. Lai, "Detection of vehicle occlusion using a generalized deformable model," in *Proc. IEEE Int. Symp. Circuits Syst.*, Jun. 1998, vol. 4, pp. 154–157.
[13] R. Cucchiara, M. Piccardi, and P. Mello, "Image analysis and rule-based reasoning for a traffic monitoring system," *IEEE Trans. Intell. Transp. Syst.*, vol. 1, no. 2, pp. 119–130, Jun. 2000.
[14] K. Otsuka and N. Mukawa, "Multiview occlusion analysis for tracking densely populated objects based on 2-D visual angles," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recog.*, Jul. 2004, vol. 1, pp. 90–97.
[15] R. P. Ramachandran, G. Arr, C. Sun, and S. G. Ritchie, "A pattern recognition and feature fusion formulation for vehicle reidentification in intelligent transportation systems," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, May 2002, vol. 4, pp. 3840–3843.
[16] B. Steux, C. Laurgeau, L. Salesse, and D. Wautier, "Fade: A vehicle detection and tracking system featuring monocular color vision and radar data fusion," in *Proc. IEEE Intell. Veh. Symp.*, Jun. 2002, vol. 2, pp. 632–639.
[17] Z. Fan, J. Zhou, D. Gao, and G. Rong, "Robust contour extraction for moving vehicle tracking," in *Proc. IEEE Int. Conf. Image Process.*, Jun. 2002, vol. 3, pp. 625–628.

[18] T. Ikeda, S. Ohnaka, and M. Mizoguchi, "Traffic measurement with a roadside vision system—Individual tracking of overlapped vehicles," in *Proc. IEEE Int. Conf. Pattern Recog.*, Aug. 1996, vol. 3, pp. 25–29.
[19] S. Kamijo, Y. Matsushita, K. Ikeuchi, and M. Sakauchi, "Occlusion robust tracking utilizing spatio–temporal Markov random field model," in *Proc. IEEE Int. Conf. Pattern Recog.*, Sep. 2000, vol. 1, pp. 140–144.
[20] O. Masoud, N. P. Papanikolopoulos, and E. Kwon, "The use of computer vision in monitoring weaving sections," *IEEE Trans. Intell. Transp. Syst.*, vol. 2, no. 1, pp. 18–25, Mar. 2001.
[21] C. C. C. Pang, W. W. L. Lam, S. X. C. He, D. C. K. Ngai, and N. H. C. Yung, "Occlusion reasoning and resolvability from a sequence of traffic images," in *Proc. IEEE Intell. Transp. Syst. Conf.*, manuscript presented and could be downloaded from IEEE Xplore, Oct. 2004.
[22] H. L. Eng, J. Wang, A. H. Kam, and W. Y. Yau, "A Bayesian framework for robust human detection and occlusion handling using human shape model," in *Proc. IEEE Int. Conf. Pattern Recog.*, Aug. 2004, vol. 2, pp. 257–260.
[23] H. Yang, J. Lou, H. Sun, W. Hu, and T. Tan, "Efficient and robust vehicle localization," in *Proc. IEEE Int. Conf. Image Process.*, Oct. 2001, vol. 2, pp. 355–358.
[24] C. Gentile, O. Camps, and M. Sznaier, "Segmentation for robust tracking in the presence of severe occlusion," *IEEE Trans. Image Process.*, vol. 13, no. 2, pp. 166–178, Feb. 2004.
[25] K. P. Lim, A. Das, and M. N. Chong, "Estimation of occlusion and dense motion fields in a bidirectional Bayesian framework," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 712–718, May 2002.
[26] D. Koller, K. Daniilidis, and H. H. Nagel, "Model-based object tracking in monocular image sequences of road traffic scenes," *Int. J. Comput. Vis.*, vol. 10, no. 3, pp. 257–281, Jun. 1993.
[27] Z. Fan, J. Zhou, D. Gao, and Z. Li, "Contour extraction and tracking of moving vehicles for traffic monitoring," in *Proc. IEEE Intell. Transp. Syst. Conf.*, Sep. 2002, pp. 84–87.
[28] A. H. S. Lai and N. H. C. Yung, "A fast and accurate scoreboard algorithm for estimating stationary backgrounds in an image sequence," in *Proc. IEEE Int. Symp. Circuits Syst.*, Jun. 1998, vol. 4, pp. 241–244.
[29] W. W. L. Lam, C. C. C. Pang, and N. H. C. Yung, "Highly accurate texture-based vehicle segmentation method," *Opt. Eng. SPIE*, vol. 43, no. 3, pp. 591–603, Mar. 2004.
[30] G. S. K. Fung, N. H. C. Yung, and G. K. H. Pang, "Camera calibration from road lane markings," *Opt. Eng. SPIE*, vol. 42, no. 10, pp. 2967–2977, Oct. 2003.
[31] A. H. S. Lai, G. S. K. Fung, and N. H. C. Yung, "Vehicle type classification from visual-based dimension estimation," in *Proc. IEEE Intell. Transp. Syst. Conf.*, Aug. 2001, pp. 201–206.



Clement Chun Cheong Pang (S'04–M'07) received the B.Eng. and M.Eng. degrees in electrical and computer engineering from McMaster University, Hamilton, ON, Canada, in 1999 and 2001, respectively, and the Ph.D. degree in electrical and electronic engineering from the University of Hong Kong (HKU), Hong Kong, in 2005.

He is currently a Senior Research Assistant with the Laboratory for Intelligent Transportation Systems Research, Department of Electrical and Electronic Engineering, HKU, and is currently working toward a unified image algebraic model for outdoor visual traffic surveillance. His research interests include the investigation of statistical models in computer vision, as well as the application of statistical models in railway signaling and communications.



William Wai Leung Lam received the B.Eng. degree in electrical engineering from McMaster University, Hamilton, ON, Canada, and the M.Phil. degree in electrical and electronic engineering from Hong Kong University of Science and Technology, Hong Kong. He is currently working toward the Ph.D. degree with the Laboratory for Intelligent Transportation Systems Research, Department of Electrical and Electronic Engineering, University of Hong Kong.

His research interests include computer vision, intelligent transportation systems, and computer networks.

Nelson Hon Ching Yung (S'82–M'85–SM'96) received the B.Sc. and Ph.D. degrees from the University of Newcastle Upon Tyne, Newcastle Upon Tyne, U.K.

From 1985 to 1990, he was a Lecturer with the University of Newcastle Upon Tyne. From 1990 to 1993, he was a Senior Research Scientist with the Department of Defence, Canberra, Australia. In late 1993, he joined the University of Hong Kong (HKU), Hong Kong, as an Associate Professor. He leads a research team in digital image processing and intelligent transportation systems. He is the founding Director of the Laboratory for Intelligent Transportation Systems Research, Department of Electrical and Electronic Engineering, HKU, and is also the Deputy Director of HKU's Institute of Transport Studies. He has coauthored five books and book chapters and has published more than 130 journal and conference papers in the areas of digital image processing, parallel algorithms, visual traffic surveillance, autonomous vehicle navigation, and learning algorithms. He is the holder of one U.S. patent and successfully transferred his research to a spin-off technology company. He regularly delivers talks/seminars to government units, professional institutions, associations and commercial companies, and gives interviews to local press regarding his research. He acts as a consultant to a number of local and international companies. He regularly serves as an expert witness for the courts of the HKSAR.

Dr. Yung is a member of the Hong Kong Institution of Engineers and the Institution of Electrical Engineers. He is a Chartered Electrical Engineer. He serves as a Reviewer for the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART C, *CASVT, Vehicular Technology, Signal Processing, The International Society for Optical Engineers (SPIE) Optical Engineering, SPIE Journal of Electronic Imaging*, *HKIE Proceedings, Microprocessors and Microsystems,* and *Robotics and Autonomous Systems Journal*. He was a member of the Advisory Panel of the ITS Strategy Review and the Transport Department, HKSAR; the Regional Secretary of the IEEE Asia-Pacific region, a Council Member and the Chairman of Standards Committee of ITS-HK; and the Chair of Computer Division of the International Institute for Critical Infrastructures. He was a Croucher Scholar, and his team received the Silver Award from the Hong Kong Electronic Industry Association for Outstanding Innovation and Technology Product (2000) for the Mobile and Online Vending EnableR (MOVER) solution. His biography has been published in Marquis' *Who's Who in the World* since 1998 and in *Who's Who in Asia*.