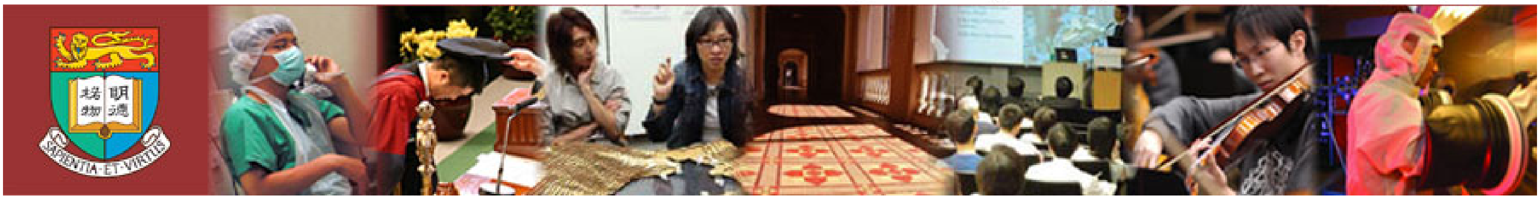


A Methodology for Determining the Resolvability of Multiple Vehicle Occlusion in a Monocular Traffic Image Sequence

Clement C. C. Pang* and Nelson H. C. Yung
Laboratory for Intelligent Transportation Systems Research
Department of Electrical and Electronic Engineering,
The University of Hong Kong, Hong Kong SAR, China

ABSTRACT

This paper proposed a knowledge-based methodology for determining the resolvability of *N* occluded vehicles seen in a monocular image sequence. The resolvability of each vehicle is determined by: firstly, deriving the relationship between the camera position and the number of vertices of a projected cuboid on the image; secondly, finding the direction of the edges of the projected cuboid in the image; and thirdly, modeling the maximum number of occluded cuboid edges of which the occluded cuboid is irresolvable. The proposed methodology has been tested rigorously on a number of real world monocular traffic image sequences that involves multiple vehicle occlusions, and is found to be able to successfully determine the number of occluded vehicles as well as the resolvability of each vehicle. We believe the proposed methodology will form the foundation for a more accurate traffic flow estimation and recognition system.

Keywords: Monocular traffic image sequence, occlusion, cubic deformable model, resolvability, curvature points.

1. INTRODUCTION

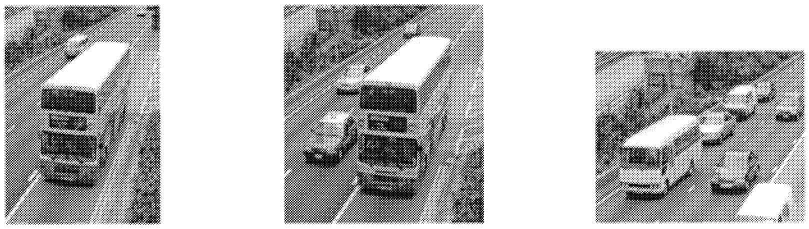
Vehicle occlusion has always been a practical yet challenging problem within the context of recognition and counting of vehicles in visual surveillance applications [1] (Fig. 1). Failing to resolve occlusion would result in (i) serious traffic flow estimation errors in congested scenarios; (ii) vehicle dimension estimation errors and (iii) recognition errors. Although a partial solution may be obtained by varying the camera mounting configuration, e.g. setting optical axis perpendicular to the road plane, it severely reduces the amount of visual information as compared with a perspective camera configuration, and limits its applications [2]. For this reason, we are motivated to find a solution to the occlusion problem based on a sequence of images captured from a standard surveillance camera. It is the purpose of this paper to present the methodology that has been developed, as well as the test results that has been obtained. The rest of this paper is organized as follow. Problem analysis is detailed in Section 2. Following that, the proposed methodology is presented in Section 3. Experimental results and discussion is depicted in Section 4. Suggestions for future development are given in Section 5, and the conclusion can be found in Section 6.

2. PROBLEM ANALYSIS

In essence, occlusion refers to the overlapping of vehicles visually in an image due to the perspective view of the camera [2]. When we tackle the occlusion problem, certain aspects have to be noted. Firstly, the size of each vehicle that involves in the occlusion might not be the same as one another (Fig. 1(a)). Secondly, a vehicle might be totally occluded by another vehicle, or totally not occluded, or in between these two boundary conditions (i.e. partially occluded). Thirdly, the number of vehicles that are involved in an occlusion event might be up to a cluster of *N* vehicles (Fig. 1(b)), or a vehicle queue (Fig. 1(c)), or even several queues in a congested situation. Fourthly, the camera viewing position might also be different when the camera undergoes pan-tilt-zoom action. We have to consider these aspects carefully when we propose a solution to the problem of occlusion.

*ccpang@eee.hku.hk; phone 852-2857-8414; fax 852-2559-8738

Applications of Digital Image Processing XXVII,
edited by Andrew G. Tescher, Proceedings of SPIE Vol. 5558
(SPIE, Bellingham, WA, 2004) · 0277-786X/04/$15 · doi: 10.1117/12.559220



(a) Different sized vehicles occlusion (b) A cluster of vehicles occlusion (c) A vehicle queue

Fig. 1. Vehicle occlusion

On the first two aspects, there is a need to consider whether a vehicle is resolvable or not from a cluster of occluded vehicles. Rules can be defined to model the extent of occlusion severity of which the vehicle is irresolvable due to the excessive lost of information. On the third aspect, there is a need to determine how many vehicles are involved within the cluster of segmented vehicles. On the fourth aspect, there is a need to derive a relationship between the camera position and the projected information of the objects on the image.

3. PROPOSED METHODOLOGY

3.1 Philosophy and Concept

In order to form a complete solution based on all the aspects outlined in the preceding section, we propose a methodology that includes a camera-vertex relationship model, a cuboid edge reasoning model, a cuboid resolvability model, and a cuboid resolving step (Fig. 2). It may be explained as first of all, it models the relationship between the number of cuboid vertices in the image and the camera position with respect to the vehicle, assuming that the occluded vehicles have been segmented from their background through some segmentation techniques [3], and that the cuboid is fitted onto the segmented vehicle (Fig. 3) [4]. Secondly, it reasons from the number of detected vertices of how many vehicles are involved in the occlusion as well as how they are occluded. Thirdly, it determines the possibility of resolving the occluded vehicle model (resolvability) into individual vehicle models from the information provided by the detected vertices. Lastly, if the resolvability index is positive, it resolves the occluded vehicle model into individual vehicle models.

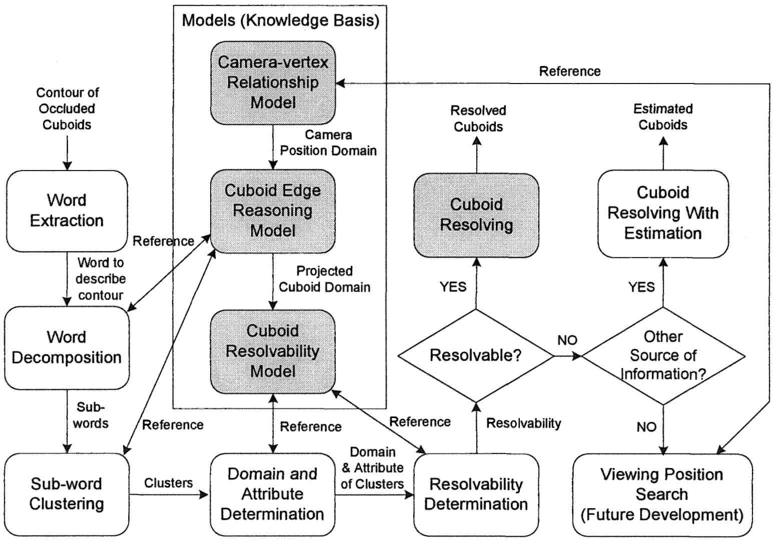

Models (Knowledge Basis)
Camera-vertex Relationship Model
Reference
Contour of Occluded Cuboids
Resolved Cuboids
Estimated Cuboids
Camera Position Domain
Word Extraction
Cuboid Edge Reasoning Model
Cuboid Resolving
Cuboid Resolving With Estimation
Reference
Word to describe contour
Projected Cuboid Domain
YES
YES
Word Decomposition
Cuboid Resolvability Model
Resolvable?
NO
Other Source of Information?
Sub-words
Reference
Reference
Reference
Resolvability
NO
Domain & Attribute of Clusters
Sub-word Clustering
Clusters
Domain and Attribute Determination
Resolvability Determination
Viewing Position Search (Future Development)

Fig. 2. Knowledge-based approach for occlusion handling

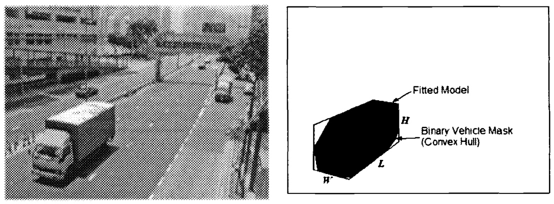
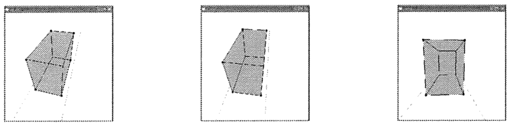


Fig. 3. Fitted cuboid model on vehicle

3.2 Camera-vertex Relationship Model

The aim of this model is to derive a relationship between the camera viewing position and the number of projected vertices of the cuboid on the image. From Fig. 4(a) to (c), it is observed that when a cuboid is projected onto the image from the 3D world coordinates to 2D image coordinates, the number of projected vertices can be 6, 5 or 4, which depends on the camera viewing position with respect to the cuboid in 3D. In order to model the number of projected vertices (i.e. n_v=6, 5 or 4), we have introduced 'virtual extended planes' on the cuboid (Fig. 5), which can be defined as the cuboid face being extended to +∞and -∞. The region of this extended cuboid surface excluding the cuboid surface itself is defined as the extended plane. Essentially, a cuboid has 8 surfaces in 3D, so there are 8 extended planes in total. The number of expected projected vertices as a function of the camera position is modeled in terms of these extended planes, which is depicted in Fig. 5. From this figure, we note that if the camera is located at the space that is immediately outside the cuboid surface, or if it is located on one of the extended planes within a range defined by the cuboid edge, then n_v=4. If the camera is located on one of the extended planes outside the range defined by the cuboid edge, then n_v=5. Otherwise, n_v=6.



(a) n_v=6 (hexagon) (b) n_v=5 (pentagon) (c)n_v=4 (tetrahedral)

Fig. 4. The three possibilities of a projected cuboid.

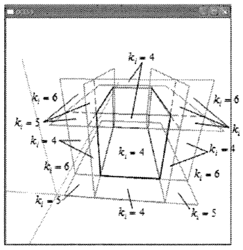


Fig. 5. The number of vertices as a function of camera

3.3 Cuboid Edge Reasoning Model

The aim of this model is to model the direction of the projected cuboid edges seen in the image with respect to the vanishing points $\mathbf{p}_y$ (i.e. parallel to the road direction), $\mathbf{p}_x$ (i.e. perpendicular to the road direction), and $\mathbf{p}_z$ (i.e. perpendicular to the ground). The set of primitives $\{a, b, c, A, B, C\}$ is used to model the direction of the projected cuboid edges, in which $\{a, A\}$ represents the lines which go into $\mathbf{p}_x$, $\{b, B\}$ represents the lines which go into $\mathbf{p}_z$, and $\{c, C\}$ represents the lines which go into $\mathbf{p}_x$ (Fig. 7). The uppercase and lowercase is used to distinguish the direction of the line when the outline of the projected cuboid is traced in a counter-clockwise manner. The bracketed set of letters represents the case when the line goes into more than one vanishing point. For example, (B, A) represents the line which goes into both $\mathbf{p}_z$ and $\mathbf{p}_y$. The set of possible projected cuboids when the camera is located in spaces defined in the previous sub-section is depicted in Fig. 7.

It is observed that the description in Fig. 7 has a general relationship. Each projected cuboid must contain the most basic set $\{b, c, B, C\}$ (Fig. 6), and we define it as 'the basis'. As it is, all the cases are formed by varying $\{a, A\}$ within the basis, which is depicted in Fig. 8.

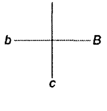


Fig. 6. Basis

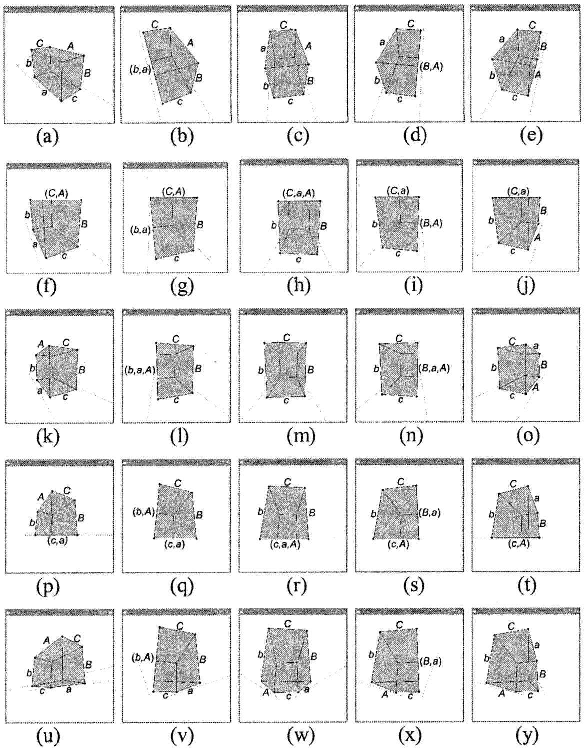


Fig. 7. The possible direction of the cuboid lines

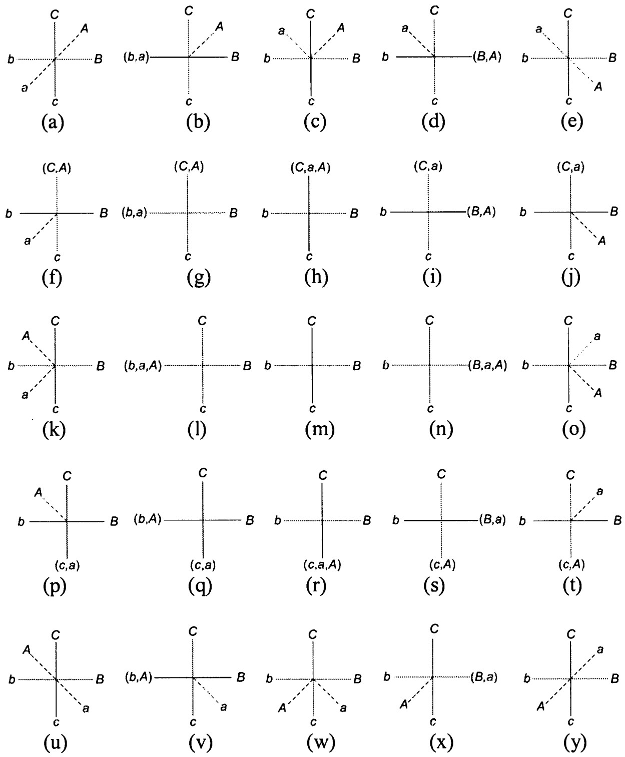


Fig. 8. The representation of the cases in terms of basis $\{b,c,B,C\}$

The representation of Fig. 8 can be easily generalized into the following 7 rules (Eqt. (1)-(7)):

1) *For n_v=4:*

Case I: $\{a, A\}$ does not exist (Fig. 8(m))
Corresponding syntax: $\{bcBC\}$ (1)

Case II: $\{a, A\}$ falls on the same basis line (Fig. 8(h), (l), (r) and (n))
Corresponding syntax (4 possibilities): $\{bcB(C,a,A)\}$, $\{(b,a,A)cBC)\}$, $\{b(c,a,A)BC\}$ & $\{bc(B,a,A)C\}$ (2)

Case III: $\{a\}$ is 90° counter-clockwise of $\{A\}$ (Fig. 8(i), (g), (q) and (s))
Corresponding syntax (4 possibilities): $\{bc(B,A)(C,a)\}$, $\{(b,a)cB(C,A)\}$, $\{(b,A)(c,a)BC\}$ & $\{b(c,A)(B,a)C\}$ (3)

2) *For n_v=5:*

Case I: $\{a\}$ is 1.5 quadrants counter-clockwise of $\{A\}$ (Fig. 8(b), (p), (x) and (j))
Corresponding syntax (4 possibilities): $\{(b,a)cBAC\}$, $\{b(c,a)BCA\}$, $\{bAc(B,a)C\}$ & $\{bcAB(C,a)\}$ (4)

Case II: $\{A\}$ is 1.5 quadrants clockwise of $\{a\}$ (Fig. 8(t), (d), (f) and (v))
Corresponding syntax (4 possibilities): $\{b(c,A)\mathrm{BaC}\}$, $\{bc(B,A)Ca)\}$, $\{bacB\,(C,A)\}$ & $\{(b,A)caBC\}$ (5)

3) *For n_v=6:*

Case I: $\{a\}$ is in opposite quadrant of $\{A\}$ (Fig. 8(a), (u), (y) and (e))
Corresponding syntax (4 possibilities): $\{bacBAC\}$, $\{bcaBCA\}$, $\{bAcBaC\}$ & $\{bcABCa\}$ (6)

Case II: $\{a\}$ is in the neighbour counter-clockwise quadrant of $\{A\}$ (Fig. 8(c), (k), (w) and (o))
Corresponding syntax (4 possibilities): $\{bcBACa\}$, $\{bacBCA\}$, $\{bAcaBC\}$ & $\{bcABaC\}$ (7)

3.4 Cuboid Resolvability Model

The aim of this model is to determine the maximum number of cuboid edges that is permitted to be completely occluded such that the missing edges of the cuboid are unrecovable. When there is occlusion, some of the cuboid lines in the image will be partially occluded, while some of them will be fully occluded, and some of them will be fully present. This can be modeled by the set *Attribute*={*F*, *P*, *L*}, where {*F*} represents 'line fully present', {*P*} represents 'line partially occluded' and {*L*} represents 'line completely occluded' (Fig. 9). Moreover, when referring to Fig. 7, one could discover that the shape of the projected cuboid can be generally modeled by the four shapes as depicted in Fig. 10 and are defined here as 'domain'. Fig. 10(a) depicts the domain for tetrahedral (D_4). Fig. 10(b) depicts the domain for pentagon (D_5), and Fig. 10(c) depicts the domain for the two possible hexagonal domains ($D_{6,1}$ and $D_{6,2}$). The set *Attribute*={*F*, *P*, *L*} could happen on every line in the domain (Fig. 10(a) to (c)).

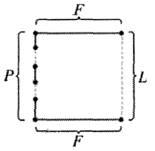
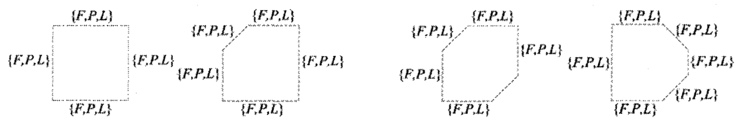


Fig. 9. A full line {*F*}, partial line {*P*} and a lost line {*L*}



(a) D_4 (b) D_5 (c) $D_{6,1}$ and $D_{6,2}$

Fig. 10. All possible outcomes of {*F*, *P*, *L*} in the D_4, D_5, $D_{6,1}$ and $D_{6,2}$ domain.

In order to derive rules to model the resolvability, we have generated all the possible combination of *Attribute*={*F*, *P*, *L*} in D_4, D_5, $D_{6,1}$ and $D_{6,2}$, and attempted to complete the missing lines for the generated samples. We have come up with the following set of rules:

```
In $D_4$,
    IF $(n_L = 0)$, ($n_L \in I$ is the number of losing lines)
        THEN resolvable,
    ELSE irresolvable.                                          (8)

In $D_5$,
    IF $(n_L = 0)$,
        THEN resolvable,
    ELSE irresolvable.                                          (9)

In $D_{6,1}$,
    IF $(n_L \leq 1)$,
        THEN resolvable,
    ELSE irresolvable.                                          (10)

In $D_{6,2}$,
    IF $\{a, A\}$ is including either $\{b\}$ or $\{B\}$,
        IF either $\{b\}$ or $\{B\}$ is lost,
            THEN irresolvable.
        ELSE IF both $\{b\}$ and $\{B\}$ exists AND $(n_L > 1)$,
            THEN irresolvable.
        ELSE resolvable.
    ELSE IF $\{a, A\}$ is including either $\{c\}$ or $\{C\}$,
        IF either $\{c\}$ or $\{C\}$ is lost,
            THEN irresolvable.
        ELSE IF both $\{c\}$ and $\{C\}$ exists AND $(n_L > 1)$,
            THEN irresolvable.
        ELSE resolvable.                                        (11)
```

3.5 Resolving Cuboids

Once a cuboid is declared as resolvable, it is resolved by completing the missing lines based on the information provided by the existing lines, so that it has one-to-one relationships with the vehicles (Fig. 11) [2].

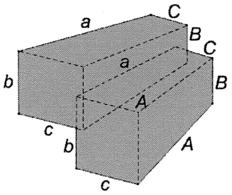
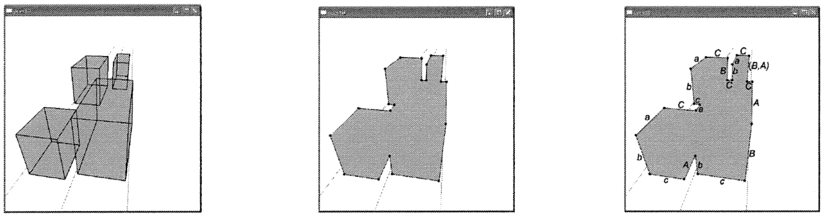


Fig. 11. Resolving cuboids

The steps to apply the proposed methodology to resolve a cluster of occluded cuboids are illustrated by the following two examples (please refer Fig. 2 for the block diagram):

(Example 1)

The first example contains a cluster of occluded cuboids (Fig. 12(a)). The outline of the cluster is shown in Fig. 12(b). The task is to resolve the clustered model into individual models if it is resolvable.



(a) A cluster of occluded cuboids (b) Contour (c) Word extraction

Fig. 12. Cuboid resolving example

1) Word Extraction (Fig. 12(c))

First of all, the outline of the cluster (i.e. contour) is reasoned by *Primitives*={*a*, *b*, *c*, *A*, *B*, *C*} based on whether the line intersects $\mathbf{p}_x$, $\mathbf{p}_y$ or $\mathbf{p}_z$. The corresponding word to represent the contour is

Word={*bcAbcBAC*(*B*,*A*)*CabCBCabcaCa*}

2) Word Decomposition

The extracted word is then decomposed into sub-words according to Eqt.(1)-(7). Once a sub-word (i.e. *Word_i*) is segmented from *Word*, the rest of the word (i.e. *Residue*) is further segmented into sub-words until *Residue*={Ø}.

1) *Word_1*={*CabcA*}	*Residue*={*bcBAC*(*B*,*A*)*CabCBCabca*}
2) *Word_2*={*bcBAC*}	*Residue*={(*B*,*A*)*CabCBCabca*}
3) *Word_3*={(*B*,*A*)*Cab*}	*Residue*={*CBCabca*}
4) *Word_4*={*C*}	*Residue*={*BCabca*}
5) *Word_5*={*BCabc*}	*Residue*={*a*}
6) *Word_6*={*a*}	*Residue*={Ø}

3) Sub-word Clustering

The segmented sub-words are then paired with each other to see if it has correlation with another sub-word, according to Eqt.(1)-(7). The result is *Word_2*, *Word_4*, and *Word_6* have correlation with each other. Therefore they are unified to form *Cluster_2*. This results in four clusters:

Cluster_1={*CabcA*}	*Cluster_2*={*bcBAC*} ∪ {*C*} ∪ {*a*}
Cluster_3={(*B*,*A*)*Cab*}	*Cluster_4*={*BCabc*}

4) Domain and Attribute Determination

Referring to Eqt.(1)-(7), domain and attribute of each cluster are given as follow:

Domain_Cluster_1 = $D_{6,1}$	*Attribute_Cluster_1*= {*FFPLPF*}
Domain_Cluster_2 = $D_{6,2}$	*Attribute_Cluster_2*= {*PFFFPP*}
Domain_Cluster_3 = D_5	*Attribute_Cluster_3*= {*PLPFF*}
Domain_Cluster_4 = $D_{6,1}$	*Attribute_Cluster_4*= {*FPLPFF*}

5) Resolvability Determination

Using Eqt.(8)-(11), the resolvability of each cluster can be determined as follow:

For *Cluster_1*, $n_L = 1\ (n_L \le 1)$ in $D_{6,1}$, ∴ Resolvable (see Eqt. 10)

For *Cluster_2*, $\{a, A\}$ is including $\{C\}$, both $\{c\}$ and $\{C\}$ exists, and other than that $n_L = 0\ (n_L \le 1)$ in $D_{6,2}$, ∴ Resolvable (see Eqt. 11)

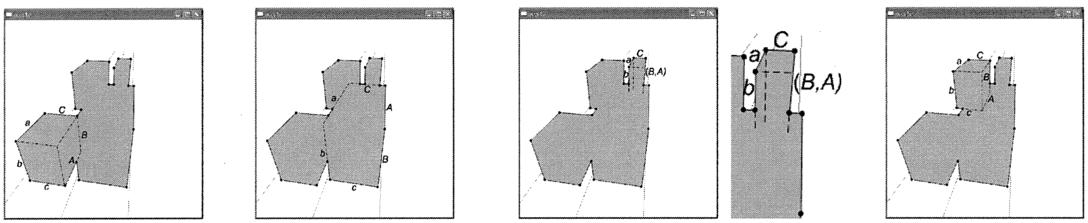
For *Cluster_3*, $n_L = 1\ (n_L \ne 0)$ in D_5, ∴ Irresolvable (see Eqt. 9)

For *Cluster_4*, $n_L = 1\ (n_L \le 1)$ in $D_{6,1}$, ∴ Resolvable (see Eqt. 10)

6) Resolving Cuboids

For the resolvable clusters, missing lines are completed in order to resolve the original cuboid shape. After completion, the cuboids are found as follow (Fig. 13):

Cuboid_1 = {*bcABCa*}	*Cuboid_2* = {*bcBACa*}
Cuboid_3 = {*b*...(*B,A*)*Ca*}	*Cuboid_4* = {*bcABCa*}



(a) Cuboid_1 (b) Cuboid_2 (c) Cuboid_3 and close-up (d) Cuboid_4

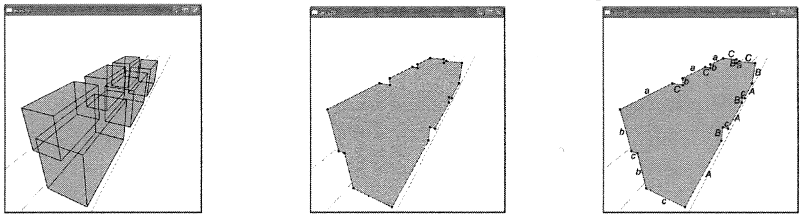
Fig. 13. Resolved cuboids

7) Resolving Cuboids with Estimation

It is noted that *Cuboid_3* is irresolvable (Fig. 13(c)). In order to determine the line $\{c\}$, the ground information has to be exploited. Since it is illogical for a vehicle to be embedded inside the ground, the two end points of line $\{c\}$ must lie on the ground. They could be found by performing a search along line $\{b\}$ and $\{(B,A)\}$ to search for the point which touches the ground, by transforming the candidate point from 2D image coordinates to 3D world coordinates to see if the point is on the ground or not[4].

(Example 2)

A more challenging example is presented here. Fig. 14(a) depicts two side-by-side queues of vehicles. Again, the task is to resolve the individual models if it is resolvable. The outline of the queue is shown in Fig. 14(b).



(a) Two side-by-side queues cuboids (b) Contour (c) Word extraction

Fig. 14. Cuboid resolving example

1) Word Extraction (Fig. 14(c))

The extracted word is *Word*={*bcbcABcABcABCaBCabCabCa*}.

2) Word Decomposition

With reference to Eqt.(1)-(7), the extracted word is decomposed into sub-words as follow:

1) $Word\_1=\{Cabc\}$	$Residue=\{bcABcABcABCaBCabCab\}$
2) $Word\_2=\{bcAB\}$	$Residue=\{cABcABCaBCabCab\}$
3) $Word\_3=\{cAB\}$	$Residue=\{cABCaBCabCab\}$
4) $Word\_4=\{cABCa\}$	$Residue=\{BCabCab\}$
5) $Word\_5=\{BCab\}$	$Residue=\{Cab\}$
6) $Word\_6=\{Cab\}$	$Residue=\{\emptyset\}$

3) Sub-word Clustering

Since none of the sub-words have correlation with each other, the resulted clusters are the same as the sub-words.

$Cluster\_1=\{Cabc\}$ $Cluster\_2=\{bcAB\}$ $Cluster\_3=\{cAB\}$ $Cluster\_4=\{cABCa\}$
$Cluster\_5=\{BCab\}$ $Cluster\_6=\{Cab\}$

4) Domain and Attribute Determination

Referring to Eqt.(1)-(7), domain and attribute of each cluster are given as follow:

$Domain\_Cluster\_1 = D_{6,1}$ or $D_{6,2}$	$Attribute\_Cluster\_1= \{FPLLPF\}$
$Domain\_Cluster\_2 = D_{6,1}$ or $D_{6,2}$	$Attribute\_Cluster\_2= \{PFFPLL\}$
$Domain\_Cluster\_3 = D_{6,1}$ or $D_{6,2}$	$Attribute\_Cluster\_3= \{LPFPLL\}$
$Domain\_Cluster\_4 = D_{6,1}$	$Attribute\_Cluster\_4= \{LPFFFP\}$
$Domain\_Cluster\_5 = D_{6,1}$	$Attribute\_Cluster\_5= \{PLLPFF\}$
$Domain\_Cluster\_6 = D_{6,1}$ or $D_{6,2}$	$Attribute\_Cluster\_6= \{PLLLPF\}$

5) Resolvability Determination

Using Eqt.(8)-(11), the resolvability of each cluster can be determined as follow:

For $Cluster\_1$, $n_L = 2\ (n_L > 1)$ in $D_{6,1}$, $\therefore$ Irresolvable in $D_{6,1}$ (see Eqt. 10).
$\{a, A\}$ is including $\{C\}$, both $\{c\}$ and $\{C\}$ exists, and other than that $n_L = 2\ (n_L > 1)$ in $D_{6,2}$,
$\therefore$ Irresolvable in $D_{6,2}$ (see Eqt. 11).

For $Cluster\_2$, $n_L = 2\ (n_L > 1)$ in $D_{6,1}$, $\therefore$ Irresolvable in $D_{6,1}$ (see Eqt. 10).
$\{a, A\}$ is including $\{B\}$, both $\{b\}$ and $\{B\}$ exists, and other than that $n_L = 2\ (n_L > 1)$ in $D_{6,2}$,
$\therefore$ Irresolvable in $D_{6,2}$ (see Eqt. 11).

For $Cluster\_3$, $n_L = 3\ (n_L > 1)$ in $D_{6,1}$, $\therefore$ Irresolvable in $D_{6,1}$ (see Eqt. 10).
$\{a, A\}$ is including $\{B\}$, but $\{b\}$ is lost, $\therefore$ Irresolvable in $D_{6,2}$ (see Eqt. 11).

For $Cluster\_4$, $n_L = 1\ (n_L \le 1)$ in $D_{6,1}$, $\therefore$ Resolvable (see Eqt. 10)

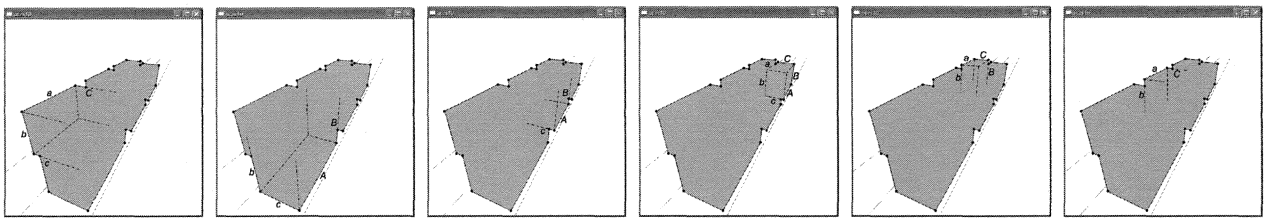
For $Cluster\_5$, $n_L = 2\ (n_L > 1)$ in $D_{6,1}$, $\therefore$ Irresolvable (see Eqt. 10)

For $Cluster\_6$, $n_L = 3\ (n_L > 1)$ in $D_{6,1}$, $\therefore$ Irresolvable (see Eqt. 10)
$\{a, A\}$ is including $\{C\}$, but $\{c\}$ is lost, $\therefore$ Irresolvable in $D_{6,2}$ (see Eqt. 11).

6) Resolving Cuboids

For the resolvable clusters, missing lines are completed in order to resolve the original cuboid shape. After completion, the cuboids are found as follow (Fig. 15):

$Cuboid\_1 = \{bc\ldots Ca\}$ $Cuboid\_2 = \{bcAB\ldots\}$ $Cuboid\_3 = \{\ldots cAB\ldots\}$
$Cuboid\_4 = \{bcABCa\}$ $Cuboid\_5 = \{b\ldots BCa\}$ $Cuboid\_6 = \{b\ldots Ca\}$



(a) *Cuboid_1* (b) *Cuboid_2* (c) *Cuboid_3* (d) *Cuboid_4* (e) *Cuboid_5* (f) *Cuboid_6*

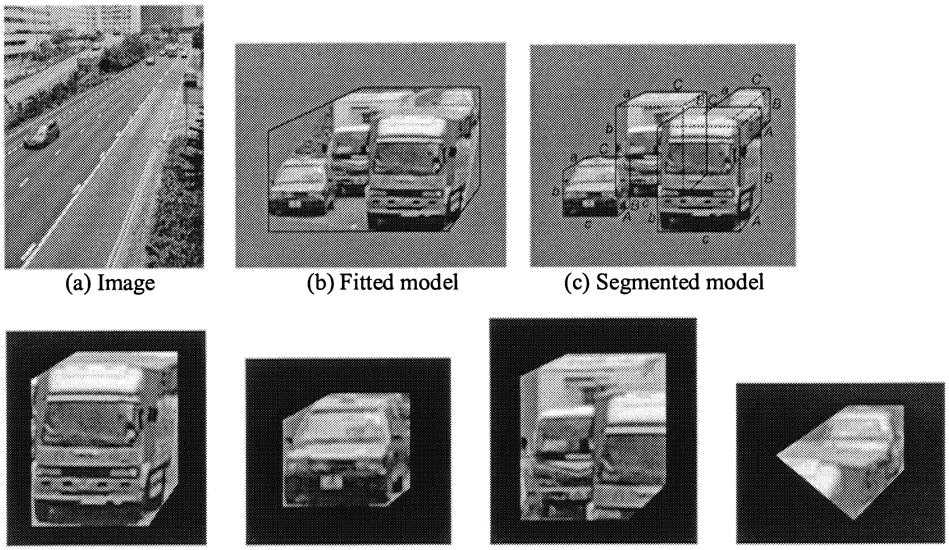
Fig. 15. Resolved cuboids

7) Resolving Cuboids with Estimation

We could observe from Fig. 15 that only *Cuboid_4* is resolvable. For the other cuboids, due to the severe lost of geometric information as a result of queued vehicles, these cuboids are irresolvable. However, the methodology has successfully identified and counted them. It is suggested that the lane width information could be exploited to approximate the shape of *Cuboid_1* and *Cuboid_3*, which could be imagined as chopping the two vehicle queues along the road direction. For *Cuboid_5*, the ground information could be exploited to determine the missing edges of the cuboid (similar to *Cuboid_3* in the previous example), and for *Cuboid_6*, both the lane width and the ground information could be used. Further researches have to be done in order to resolve *Cuboid_2*, as there is no reference in the sky to determine the top edge of the cuboid.

4. RESULTS AND DISCUSSIONS

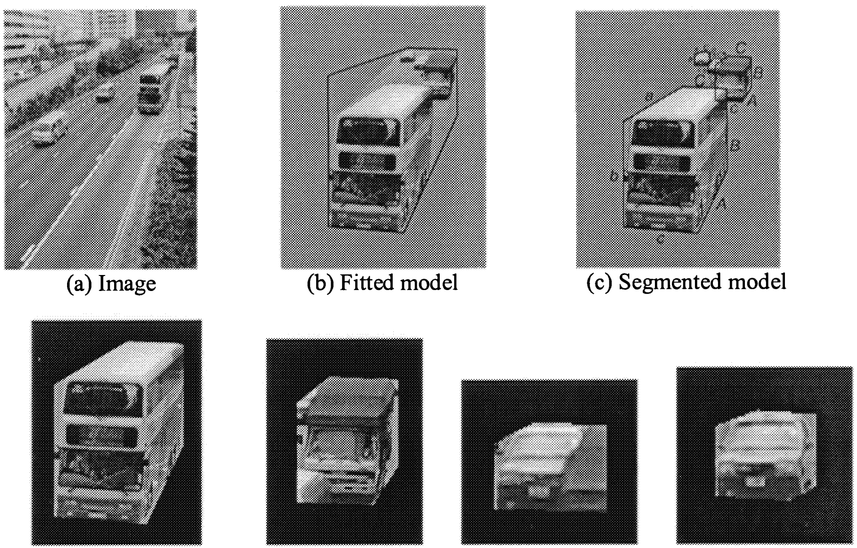
The proposed methodology has been tested rigorously on real world traffic images that contain multiple different sized vehicles that are occluded together. The first set of image contains a cluster of vehicles far away from the camera, including two trucks and two taxis. The model that is fitted onto the cluster of vehicles is shown in Fig. 16(b). The proposed methodology successfully reasoned the vertices on the outline of the segmented cluster of vehicles and recovered the individual models of the two trucks and the front taxi (Fig. 16(c)), while the back taxi is declared as irresolvable. The close-up views of the segmented vehicles are depicted in Fig. 16(d) to (g). We observe from the segmented vehicles that the recovered models fit onto the two trucks as well as the front taxi well.



(a) Image (b) Fitted model (c) Segmented model

(d) Right truck (e) Front taxi (f) Left truck (g) Back taxi

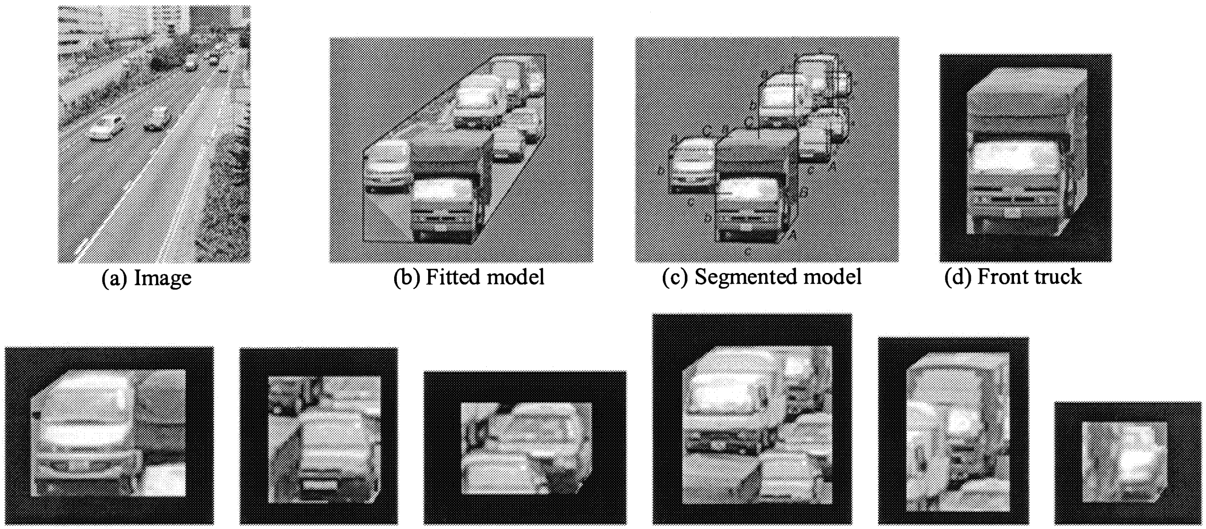
Fig. 16. Two trucks and two taxis



(a) Image (b) Fitted model (c) Segmented model

(d) Bus (e) Truck (f) Right taxi (g) Left taxi

Fig. 17. Double-decker bus, truck and two taxis



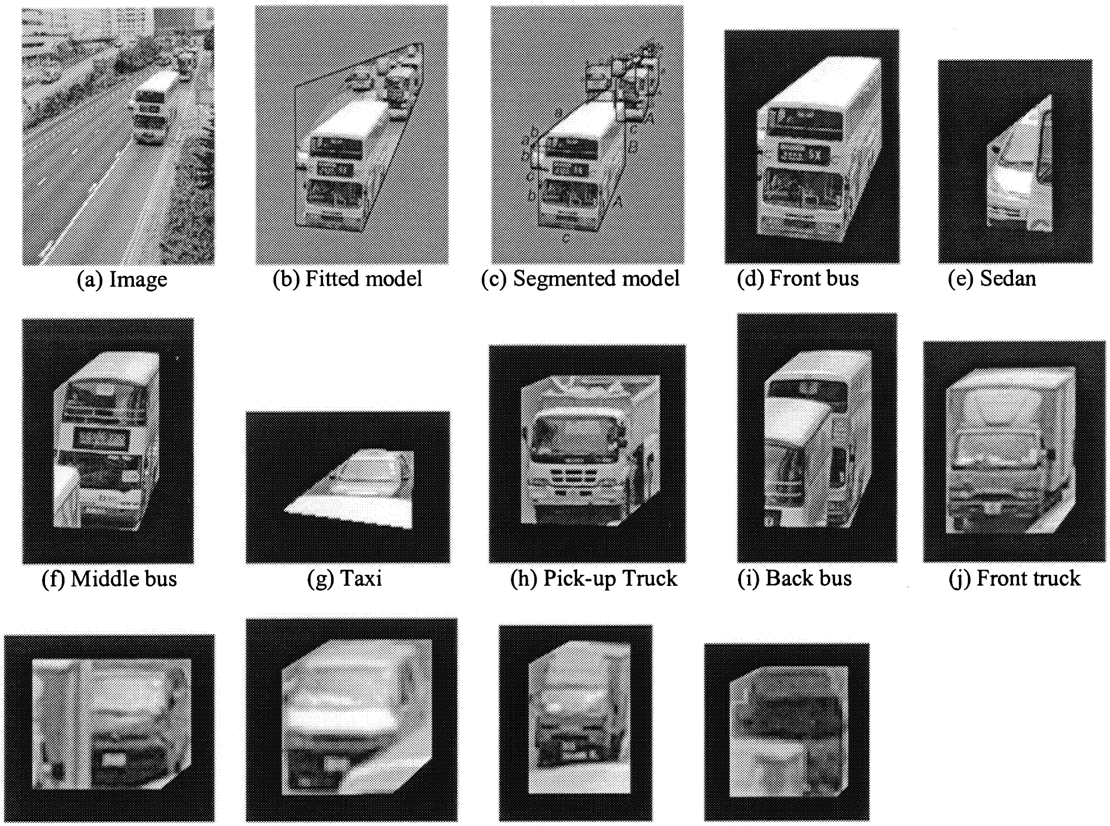
(a) Image (b) Fitted model (c) Segmented model (d) Front truck

(e) Mini-van (e) Front Sedan (f) Taxi (g) Middle truck (h) Back truck (i) Back sedan

Fig. 18. Three trucks, two sedans, mini-van and taxi

The second set of image contains a double-decker bus that has occluded a truck, while the truck is occluding two taxis (Fig. 17). Once again, the proposed methodology successfully reasoned and recovered the individual models of the double-decker bus, truck and the left taxi. The right taxi is declared as irresolvable due to insufficient information available to recover the model.

The third set of image contains a relatively big cluster of vehicles, which involves three trucks, two sedans, one mini-van and one taxi (Fig. 18). The proposed methodology successfully reasoned the number of vehicles that are involved in the cluster, and computed the resolvability of each vehicle. The result is that the front truck is declared as resolvable, while the rest of vehicles are declared as irresolvable. On the attempt to resolve the occlusion, we could see that the model of the truck is accurately resolved, while the rest of the vehicles are not.

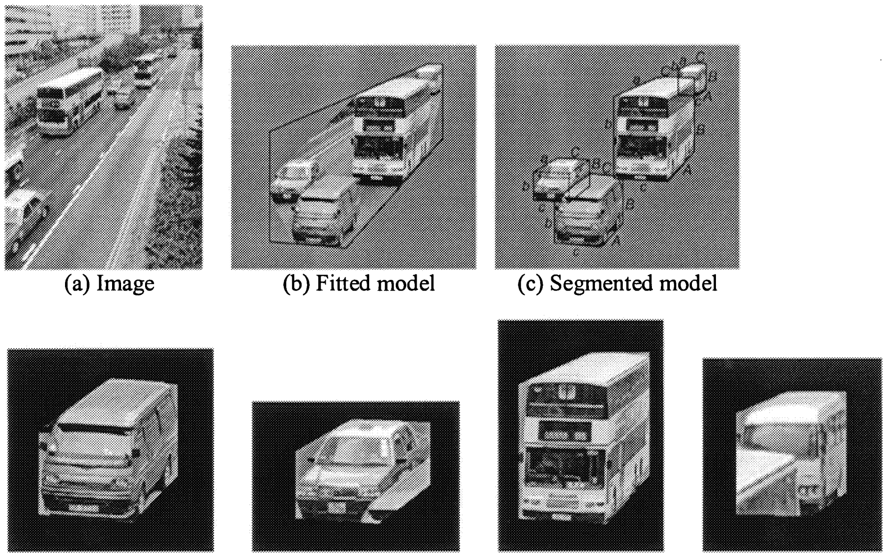


(a) Image (b) Fitted model (c) Segmented model (d) Front bus (e) Sedan

(f) Middle bus (g) Taxi (h) Pick-up Truck (i) Back bus (j) Front truck

(k) Front mini-van (l) Middle mini-van (m) Back Truck (n) Back mini-van

Fig. 19. Three double-decker buses, sedan, taxi, pick-up truck, two trucks and three mini-vans



(a) Image (b) Fitted model (c) Segmented model

(d) Mini-van (e) Taxi (f) Bus (g) Mini-bus

Fig. 20. Mini-van, taxi, double-decker bus and mini-bus

The fourth set of image contains a vehicle queue that involves three double-decker buses, a sedan, a taxi, a pick-up truck, two trucks and three mini-vans (Fig. 19). The proposed methodology successfully reasoned the number of vehicles that are involved in the queue and computed the resolvability of each vehicle. The result is that the three double-decker buses, the pick-up truck, the front truck and the middle mini-van are declared as resolvable, while the rest of the vehicles are declared as irresolvable.

The fifth set of image contains a mini-van, a taxi, a double-decker bus and a mini-bus (Fig. 20(a)). The proposed algorithm successfully reasoned and recovered the individual models of the vehicles (Fig. 20(c) to (g)).

5. SUGGESTION FOR FUTURE DEVELOPMENTS

For the proposed methodology, an immediate question raised would be on how to deal with vehicles that are irresolvable (as for the *Cuboid_3* case in *Example 1*, *Cuboid_1*, *2*, *3*, *5*, *6* in *Example 2*, and the back taxi in Fig. 16(g), etc). One possible solution is to exploit the ground information as well as the lane width information to approximate the bottom edge and side edge of the model. Recall from Fig. 2 that if the model is declared as irresolvable, then, it will immediately seek for other source of information such as the ground and the lane width to approximate the missing edges of the model. On the other hand, if such reference information is absent, as for the *Cuboid_2* case in *Example 2*, what can be done then? One possible solution is to make the camera self-adjustable so that it can search for a better view based on the information provided by the *Camera-vertex relationship model*. This opens a new avenue for future research.

6. CONCLUSION

This paper proposed a knowledge-based approach to determine the resolvability of *N* occluded vehicles seen in a monocular image sequence. A methodology is proposed to resolve the occlusion. The advantage of such approach is that it divides the occlusion problem into hierarchical knowledge basis such that the problem can be solved logically with reference to the knowledge basis. The proposed methodology has been tested on real world traffic image sequences, and has shown that it is feasible in resolving the individual model of vehicles in a vehicle cluster or queue. Future developments would be focused on how to deal with irresolvable vehicles as well as seek for boundary conditions in which the existing methodology cannot handle.

REFERENCES

1. N. H. C. Yung and A. H. S. Lai, "Detection of vehicle occlusion using a generalized deformable model", *Proceedings of the IEEE International Symposium on Circuits and Systems*, Vol. 4, pp. 154-157, 1998.
2. 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 Transactions on Intelligent Transportation Systems*, Vol. 5(3), to be published in Sep 2004.
3. W. W. L. Lam, C. C. C. Pang and N. H. C. Yung, "A highly accurate texture-based vehicle segmentation method", *Optical Engineering- The Journal of SPIE*, Vol. 43(3), pp. 591-603, 2004.
4. A. H. S. Lai, G. S. K. Fung and N. H. C. Yung, "Vehicle type classification from visual-based dimension estimation", *Proceedings of the IEEE Intelligent Transportation Systems Conference*, pp. 201-206, 2001.