International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:2, No:3, 2008

Over-Height Vehicle Detection in Low Headroom Roads Using Digital Video Processing

Vahid Khorramshahi, Alireza Behrad, and Neeraj K. Kanhere

Abstract**—**In this paper we present a new method for over-height vehicle detection in low headroom streets and highways using digital video possessing. The accuracy and the lower price comparing to present detectors like laser radars and the capability of providing extra information like speed and height measurement make this method more reliable and efficient. In this algorithm the features are selected and tracked using KLT algorithm. A blob extraction algorithm is also applied using background estimation and subtraction. Then the world coordinates of features that are inside the blobs are estimated using a noble calibration method. As, the heights of the features are calculated, we apply a threshold to select over-height features and eliminate others. The over-height features are segmented using some association criteria and grouped using an undirected graph. Then they are tracked through sequential frames. The obtained groups refer to over-height vehicles in a scene.

Keywords**—**Feature extraction, over-height vehicle detection, traffic monitoring, vehicle tracking.

I. Introduction

THERE are several systems that have been developed to help us in traffic monitoring. These systems are parts of intelligent transportation systems. Electronic road pricing, electronic toll collection system, urban traffic control system, online traffic information system and over-height vehicle detection systems are some common systems that are currently used. Over-height vehicle detection systems are warning devices which designed to alert drivers if their vehicles are over-height for an upcoming bridge, overpass, overhead walkway, etc. and make them take actions that eliminate damage to their vehicles and the structure ahead [1].

Different types of sensor technologies are used for over-height vehicle detection. One type detects the over-height vehicles by the transmission of the optical beam from transmitter to receiver. When a vehicle breaks or interrupts the beam, it is detected as an over-height vehicle. The other type, laser radar, operates in more similar manner, except that it transmits longer wavelength energy in the near-infrared spectrum [2]. The common over-height detection systems work by installing infrared beam from the transmitter to the receiver, placed directly across the lane or lanes of traffic to be monitored and an inductive loop to detect presence of vehicle in advance of the warning sign. Once the system detects an over-height vehicle by the breaking of the infrared beam, the system will activate the warning signs to instruct the driver to avoid the collision [1]. Another sensor that can be used in detecting over-height vehicle is vision sensor. The related system performs detection using a single digital camera. In this paper, we present a method for over-height vehicle detection in roads which can be used in video image processing systems. Using a camera instead of other sensors has several advantages. In addition to over-height vehicle detection, this method can provide different information from traffic like speed measure, vehicle classifying information, traffic flow rate and etc. There is no need to use loop detectors to detect vehicles as well. Comparing to laser radars, video image processing systems do not require lane closure for maintenance. The vision sensors and computer vision algorithms are used in [3-5] to calculate traffic parameters. The proposed method in [3] is based on the establishment of correspondences between regions and vehicles, as the vehicles move through the image sequence and the algorithm in [4] is based on the combination of a per pixel background model and a set of single hypothesis foreground models based on a general model of object size, position, velocity, and color distribution. The method in [5] is texture based.

In this paper, we propose a new method for over-height vehicle detection. The proposed algorithm has two phases. The first phase is done offline and the second one is online. First phase includes the calibration and detection of a cubic area called cubic detection zone. In this phase a mapping between frame coordinate and world coordinate is obtained. In other words, the goal of this offline phase is obtaining calibration matrix which is presented in section II. The calibration matrix plays a key role in our method. The second phase starts with obtaining foreground mask. The foreground mask is a binary image containing some blobs showing the moving vehicles. To obtain foreground mask, the background model is estimated and subtracted from the current frame; the related algorithms are discussed in section III. To segment vehicles and determine over-height vehicle, it is necessary to extract some feature points and track them in the block of frames. The feature extraction and tracking algorithms are

Manuscript received March 28, 2008.

V. Khorramshahi is with the Faculty of Engineering, Shahed University, Tehran, Iran (e-mail: vkhorramshahi@yahoo.com).

A. Behrad is with the Faculty of Engineering, Shahed University, Tehran, Iran (e-mail: behrad@shahed.ac.ir).

N. K. Kanhere is with the Electrical and Computer Engineering Department Clemson University, Clemson, SC (e-mail: nkanher@clemson.edu).

described in section IV. In section V, the world coordinates of the feature points inside the blobs are estimated. Then, a correction algorithm is applied to the obtained estimations. The proposed algorithm is explained in section VI. In the next step, the segmentation and grouping algorithms are applied to over-height features. The segmentation is based on some association criteria which are explained in section VII.A and the grouping problem is expressed as a graph problem in section VII.B.

II. Calibration Matrix Calculation

A. Mapping

We use a perspective-projective pinhole camera model. Calibration matrix is estimated from prerecorded videos, so it is an offline process. There is no need to know camera specification like focal length. However, we use calibration equations. The linear matrix equation of perspective projections is as follows:

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & c_{13} & c_{14} \\ c_{21} & c_{22} & c_{23} & c_{24} \\ c_{31} & c_{32} & c_{33} & c_{34} \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} . \qquad (1)$$

What is interesting about vector $[x_1,x_2,x_3]$ is that the ratios (x_1/x_3) and (x_2/x_3) are nothing but the image coordinates [7]. These ratios are referred as (x_{im},y_{im}). (X_w,Y_w,Z_w) are the coordinates in world coordinate system. A map between world coordinates and frame coordinates is obtained from (1):

$$\begin{aligned} x_{\text{im}} &= X_w c_{11} + Y_w c_{12} + Z_w c_{13} + c_{14} \\ &- x_{\text{im}} X_w c_{31} - x_{\text{im}} Y_w c_{32} - x_{\text{im}} Z_w c_{33} \end{aligned} . \qquad (2)$$

$$\begin{aligned} y_{\text{im}} &= X_w c_{21} + Y_w c_{22} + Z_w c_{23} + c_{24} \\ &- y_{\text{im}} X_w c_{31} - y_{\text{im}} Y_w c_{32} - y_{\text{im}} Z_w c_{33} \end{aligned} . \qquad (3)$$

By determining a cubic detection zone, we have eight image points in frame coordinate system and their correspondences in world coordinate system. Each correspondence yields two equations of the form (2) and (3). The calibration matrix is obtained using these equations. Six or more correspondences lead to an over-determined system which can be solved using a least mean squared technique.

B. Cubic Detection Zone

As mentioned earlier, we need at least six feature points in the frame coordinate system and their correspondences in world coordinate systems to calculate the calibration matrix. The feature points in the frame coordinate system are obtained by determining the vertexes of a cubic area called cubic detection zone. The corresponding points for the vertex of the cubic area in world coordinate system are also calculated according to known width of the road and dimensions of a known vehicle. To determine the cubic detection zone, a sample video of the traffic of a place where the camera is supposed to be installed should be recorded. Then the user should mark some points in a way which is presented in next section. The cubic detection zone can be obtained from any vehicle, but trucks and buses provide bigger zones, so it is preferred to use trucks instead of other vehicles.

The world coordinate system is user defined. So, we define a three dimensional Euclidian world coordinate system somehow that the Z_w axis be perpendicular to the road.

C. Marking Procedure

Fig. 1 shows the result of marking procedure for a truck with known dimensions. The marking procedure is as follows [6]:

a) A vehicle is selected by user (the dimensions of vehicle are known).

b) Two points (point number 1 and 2) are marked at far left and far right of the road. A line connects them to each other.

c) The video is played until the vehicle passes the line obtained from previous step.

d) Point number 3 is marked at far left of the road.

e) Point number 4 and 5 are marked at the height of vehicle at far left and far right of the road. Point number 1, 2, 4 and 5 are connected to each other to create the back side rectangle of cubic.

f) The video is played until the end of vehicle reaches the point number 3. The point number 5 and 6 are marked and connected to each other as step b. Point number a, b, f and g are connected to each other to create the down side rectangle of cubic.

g) The point number 8 and 9 are marked as step c and e. Point number 6, 7, 8 and 9 are connected to each other to create the front side rectangle of cubic. Point number 4, 5, 8 and 9 are connected to each other to create the top side rectangle of cubic.

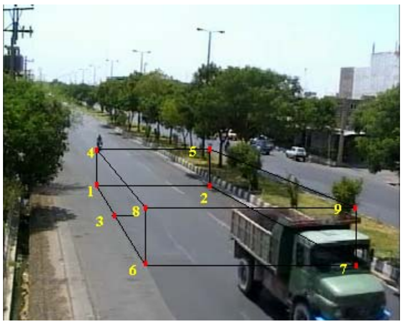

Fig. 1 The result of marking procedure for a truck with known dimensions

III. Obtaining the Foreground Mask

The Foreground mask is a binary image showing moving vehicles as blobs in the scene. To constitute foreground mask, firstly, a model of background image should be estimated. Then, the background image should be subtracted from processed scene frames to obtain the foreground mask for each frame [8]. After applying threshold, the foreground contains some blobs showing the moving vehicles in the scene. To obtain closed and noise free blobs, it is necessary to apply some morphology operations to resultant image as well. To estimate the background image model, we simply use a temporal median filter.

IV. Processing a Block of Frames

Frequent occlusions and appearance changes (as vehicles approach the camera) cause losing a large number of features, so it is preferred to process a block of frames instead of processing only two subsequent frames. To solve the problem of missing features and to get better segmentation results, we do segmentation process in two stages: the first stage is done in every frame which we called it feature segmentation and second stage is done throughout a block of F frames, containing current frame and F-1 previous frames, therefore each block overlaps with the previous block by F-1 frames. We also use Kanade-Lucas-Tomasi (KLT) algorithm [9] to extract and track features points.

V. World Coordinates Estimation of Feature Points

It is necessary to calculate the 3D coordinates of the feature points in world coordinate system, to detect over-height vehicle. Therefore, (2) and (3) are rewritten as follows:

$$\begin{bmatrix} X_w \\ Y_w \end{bmatrix} = \begin{bmatrix} c_{31}x_{\text{im}} - c_{11} & c_{32}x_{\text{im}} - c_{12} \\ c_{31}y_{\text{im}} - c_{21} & c_{32}x_{\text{im}} - c_{22} \end{bmatrix}^{-1} \times \begin{bmatrix} c_{14} - x_{\text{im}} + Z_w(c_{13} - c_{33}x_{\text{im}}) \\ c_{24} - y_{\text{im}} + Z_w(c_{23} - c_{33}y_{\text{im}}) \end{bmatrix}. \qquad (4)$$

As we mentioned earlier, we define a three dimensional Euclidian world coordinate system somehow that the Z axis be perpendicular to the road. The Z axis refers to the height and the origin of the coordinates system is placed on the road. To estimate the world coordinates of a feature point, first, we project the feature point using blobs on the road vertically as shown in Fig. 2. So, the vertical projection of any feature point in blob is the last pixel of the blob in vertical direction, exactly under the feature point. Therefor, the feature point and its projection have same x but different y in 2D coordinates. They also have same X_w and Y_w but different Z_w (because of flat road assumption) in 3D coordinates.

As explained, the road is supposed to be flat, so the height of projected point would be zero. Then with calculated calibration matrix and known projected point coordinates, the world coordinates (X_w,Y_w,0) for projected point can be calculated using equation (4). Therefore, the projected coordinates in world coordinates is known, then according to the flat road, the only difference between feature point and its projection is the height. Using equations (2) and (3), the height of each feature point can be calculated.

In this step of algorithm, the calculation of heights of feature points can not be reliable due to shadows and occlusion situations. So, the calculated heights need to be corrected as we explain it in section VI. By the way, the features in the blobs are the only useful features which are participated in segmentation algorithm.

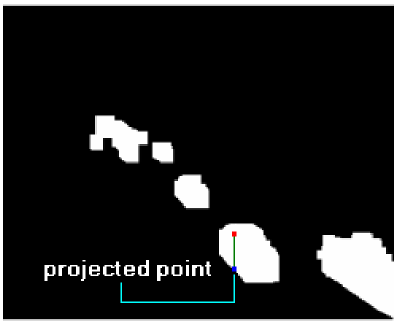

Fig. 2 Vertical projection of a feature point on the road using blobs in 2D coordinates

VI. Height Correction

In previous step, height estimations of some feature points are not reasonable in world coordinate system. This occurs when occlusion happens. The shadow of the vehicles on the road is another reason which results in irrational heights. We divide the results of height estimation algorithm into two groups; reasonable and unreasonable heights. Reasonable heights are related to features, which their heights are less than a user defined threshold. Other estimations higher than this threshold are unreasonable and need to be corrected. Because of the rigid motion of vehicles and with respect to representation of a line in world coordinate as (5), correction is done using reasonable height estimations.

$$P = P_R + \alpha[P_R - P_H] \ . \qquad (5)$$

Where P_H and P_R are 3 x 1 vectors representing any two points on the line, and α is a scalar which defines location of a point along the line. Here, we select P_H as the projection of P at the top side of the cubic zone and P_R as the projection of point P at the down side of it. As shown in Fig. 3, we consider two points; P and Q which undergo a translational motion from P^0, Q^0 in first frame of block to P^t, Q^t after duration t. So, P^t - P^0 would be equal to Q^t - Q^0. α is derived using (5) and the mentioned equality as:

$$\alpha = \frac{[Q^t - Q^0] - [P_R^t - P_R^0]}{[P_H^t - P_H^0] - [P_R^t - P_R^0]} \ . \qquad (6)$$

In any frames, there are some feature points with reasonable height estimation. We use them to correct unreasonable height estimation. So, there would be several corrections according to the reasonable height estimation for any feature points with unreasonable height estimation. We derive (7) by substituting (6) in (5):

$$P_j = P_R + \frac{[Q_j^t - Q_j^0] - [P_R^t - P_R^0]}{[P_H^t - P_H^0] - [P_R^t - P_R^0]}[P_R - P_H] \ . \quad (7)$$

$1 \leq j \leq$ number of reasonable height estimations

Among all the corrections, we select the correction that minimizes the weighted sum of Euclidean distance and squared trajectory error over all F frames in the block. The Euclidean distance is calculated in the last frame of each block between each feature point with unreasonable height estimation and all feature points with reasonable height estimation. The trajectory error is considered from the first frame to the last frame of each block for each feature point with unreasonable height estimation and all feature points with reasonable height estimation.

In the next step, after height correction, we only keep the feature points which are over a user defined value and eliminate the feature points under that.

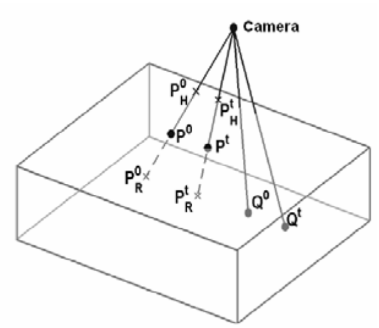

Fig. 3 P has unreasonable height. Q has reasonable height

VII. Grouping and Tracking

A. Feature Segmentation

The reason for feature segmentation is the grouping of the features in meaningful groups which refer to different vehicles. As mentioned before, we do segmentation process in two stages. The first stage is done in every frame which we call it feature segmentation. We have proposed a new method which can efficiently segment features to meaningful groups. Segmentation problem is supposed to be an undirected weighted graph problem. Features are nodes of the graph and the weight of an edge connecting the respective nodes are the measures of their association to each other. The association criteria, which we have used, are the combination of Euclidian distance criterion, trajectory error criterion and background measure criterion. The Euclidian distance is the Euclidian distance between each pair of features at the last frame of each block. Trajectory error is the magnitude of difference between the displacements of each pair of features throughout a block. The background measure criterion is the number of background pixels between each pair of features. Nodes with approximate edge weight values can be grouped as a single group. The association threshold value (or edge weight value) is selected by user and depends on the situation of traffic. This primarily basic grouping may not result in meaningful results, so we define meaningful groups. A group is meaningful if it has a minimum number of required features and the centroid of features is inside the detection zone in X_w and Y_w directions.

B. Tracking Meaningful Groups

This step is considered as the second stage of grouping. We constitute a block of F frames for each frame, overlapping with the previous block by F-1 frames. In the previous section we obtained some meaningful groups for every block; however these groups are not our final groups. To have better results, it is necessary to apply some other processes to obtained meaningful groups. To do this, we track the meaningful groups of previous section and split and merge them using the information of consecutive frames. Therefore, it is necessary to find correspondence between groups within consecutive blocks. To track these groups over consecutive blocks; we use an undirected weighted graph which relates the groups of previous block to current block [6]. The nodes of this graph are meaningful groups of previous and current blocks and edge weights are the number of common features shared by these groups. A group in previous block sharing features with more than one group from current block indicates splitting. In this case an edge with maximum weight is kept and others are eliminated. Two or more groups in previous block sharing common features with a group in current block indicate merging. In this case an edge with maximum weight is kept and others are eliminated. A group in previous block having no association is considered as a missing event, and a group in current block having no association with any of the groups in the previous block is considered as a new detection. If a group in the previous block shares features with only a single group in the current block, then we call this a unique correspondence. If a group has a predetermined number of unique correspondences over several blocks, we call it reliable group which indicates an over-height vehicle.

VIII. Experimental Results

The proposed algorithms were implemented in Visual C++ on a Pentium IV, 3.4 GHz platform using Intel OpenCV image library. The algorithm was tested on three different videos captured at 30 frames per second and digitized at 400 × 296 resolutions. The original videos were color videos; however the algorithms needs only gray level information of the videos so we converted the color videos to gray scale. We do not perform any preprocessing. The videos contains different situation from low occlusion to severe occlusion. Cubic detection zone determination was done using trucks, separately.

The vehicles over 2.5 meters are detected as over-height vehicles in all videos. First video was recoded from side of the road. The camera was placed on an approximately 6 meters pole. The over-height vehicles are shown in Fig. 4 for the first test. Fig. 5 shows that the algorithm is also capable of detecting over-height vehicles in opposite way. Second video was recorded from a pedestrian overpass with about 10 meters height. You can see an example of over-height vehicle detection for second test in Fig. 6.a.

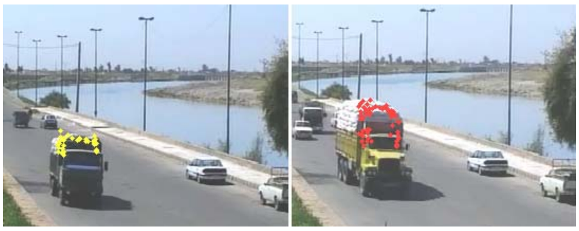

Fig. 4 Result of first test, an over-height vehicle detection (over 2.5 meters)

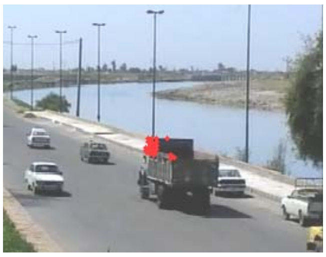

Fig. 5 The ability of algorithm in detecting over-height vehicles in opposite ways

Third video was recoded from side of the road. The camera was placed on an approximately 9 meters pole. It was recorded in a windy day, which made the trees a big problem. As the leaves moves, they are detected as moving blobs, therefore we eliminated the features which were out of the range of road way and that made the result of algorithm more accurate.

It is important to note that the proposed algorithm not only is able to find the over-height vehicles in the road, but also the proper segmentation and grouping algorithms provide us to segment the vehicles in the case of sever occlusion as shown in Fig. 6.b. This enables us to provide other traffic information such as the count of over-height vehicles correctly (useful for traffic engineering).

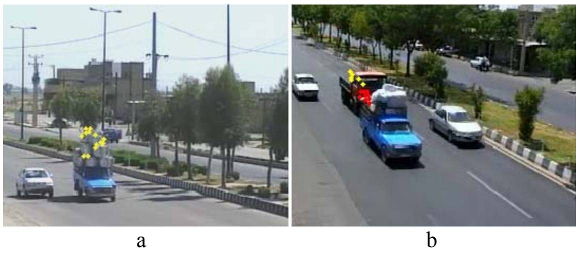

Fig. 6 a. Result of second test, an over-height vehicle detection (over 2.5 meters) b. Result of third test, an over-height (2.5 meter) detection in occlusion situation

A test was also done for over-height vehicle counting for all videos with earlier mentioned specification. The result of the algorithm is shown in Table I. As it is shown in Table I, the results are satisfactory. The experimental results showed that, the proposed algorithm can be used in automatic warning sign systems to caution the drivers and lead them to other directions.

TABLE I
RESULT FOR OVER-HEIGHT (OVER 2.5 METERS) VEHICLE COUNTING FOR THREE DIFFERENT VIDEOS

	Real number of over-height vehicles	Number of detected over-height vehicles by algorithm	Length of video
Test 1	15	13	20 minutes
Test 2	5	5	13 minutes
Test 3	10	9	6 minutes

IX. CONCLUSION

Comparing to optical systems and laser radars, this method provides some extra information which is really useful. Less maintenance problems and lower price are the other advantages of presented algorithm. For better feature segmentation, we are working on possible association criterion like color. Some preprocessing may help in more accurate height estimations, too. Also we are studying the noise problem because it can be important as a perturbation to detect the feature points.

ACKNOWLEDGMENT

We would like to thank Mr. Nader Khorramshahi for assisting us in recording different videos.

REFERENCES

[1] CSE Intelligent Transport Systems, online: http://www.cse-global.com/solutions/its_product.asp

[2] L. A. Klein, M. K. Mills and D. R. Gibson, *Traffic Detector Handbook*. vol. 1, Federal Highway Administration, 2006.

[3] S. Gupte, O. Masoud, R. F. K. Martin, and N. P. Papanikolopoulos, “Detection and classification of vehicles”, *IEEE Trans. on Intelligent Transportation Systems*, vol. 3, pp. 37-47, Mar. 2002.

[4] D. Magee, "Tracking multiple vehicles using foreground, background and motion models", *Image and Vision Computing,* vol 22(2), pp. 143-155, 2004.

[5] M. Haag and H. Nagel, "Combination of edge element and optical flow estimate for 3D-model-based vehicle tracking in traffic image sequences", *International Journal of Computer Vision,* pp. 295-319, 1999.

[6] N. Kanhere, S. Pundlik and S. Birchfield, “Vehicle segmentation and tracking from a low-angle off-axis camera”, in *IEEE Conference on ComputerVision and Pattern Recognition*, pp. 1152-1157, 2005.

[7] E. Trucco and A. Verri, *Introductory Techniques for 3-D Computer Vision*. Prentice Hall, 1998, pp. 39.

[8] X. Fu, Z. Wang, D. Liang and J. Jiang, “ The Extraction of Moving Object in Real-Time Web-Based Video Sequence”, in *The 8th International Conference on Digital Object Identifier,* Vol. 1, pp. 187-190, 2004.

[9] C. Tomasi and T. Kanade, ”Detection and tracking of point features,” Carnegie Mellon University , Technical Report CMU-CS-91-132, 1991.