Journal of Engineering and Applied Sciences 12 (8): 2207-2213, 2017
ISSN: 1816-949X

Realtime Stereo Vision for Vehicle Detection, Classification and Counting Using Raspberry Pi

Mohammad Iqbal, Rudi Trisno Yuwono,
Hadyan Mardhi Fadlillah and Sigit Widiyanto
Department of Computer Engineering, Faculty of Computer Science and Information Technology,
Gunadarma University, Depok, Indonesia

Abstract: In Indonesian toll road system, still found the lack of information on the number of vehicles passing through to the road in realtime. This is caused by the absence of detection and vehicle counting system that work in realtime applied on the road toll and this situation can cause difficulties to controll the traffic on the toll road. Therefore, it necessary to study an automated system that works in realtime doing precisely identifying the type of vehicle and calculate it. In this research, we built a prototypes of visual based vehicle detection, classification and counting, made using mini PC raspberry Pi as the central processing and USB camera modules as input devices and arrange in Stereo System to reduce the inability to detect vehicles behind another vehicle. Some algorithms of computer vision assembled from the functions that exist in the library openCV. For realtime segmentation method using Haar-like features, then we uses that found features as reference from every stereo images and apply the ratio test to find the best matches and extract the locations of matched keypoints in both the images. RANSAC algorithm is used to minimize errors that occur after matching. So, best matches which provide correct estimation (inliers) and throw out remaining outliers. The results showed improvements of vehicles that can be detected and counted.

Key words: Vehicle detection, classification and counting, intelligent transportation system, openCV, raspberry Pi

INTRODUCTION

Volume of vehicles information passing on the toll road is needed by operator of toll roads, transportation and traffic department and also for local governments. This data widely used for calculating the density of vehicles, counting the frequency of vehicles, calculating average pollution can also be used as a reference for the improvement of existing roads, widening of the road, new roads development and the arrangement of traffic signs, etc. The function of toll road is not only a freeway but also as a gateway between cities and provinces. Uncontrolled flow of vehicle traffic can result in severe congestion along the road. In order for tolls to operate optimally, the flow of the traffic needs to be controlled intensively. One of the possible way to do this is by achieving information of the number of vehicles on the highway. To address this problem, we need a system that can automatically classify classes of vehicle and calculate the number of vehicles entering and exiting in toll road.

MATERIALS AND METHODS

The research method used in this study is experimental methods to build a prototype Intelligent Transportation System (ITS) visual based vehicle detection, classification and counting. It started with the literature study, collecting data from various sources such as annual report book toll road traffic taken from the web site Indonesian toll roads operators as well as data from previous research studies to be used as a guide or reference this research. The next step is doing an observation about functionality and the use of devices or tools associated in visual based method such as camera and many visual algorithm that will be used on a system. Then, we built the prototypes based-on design and at the end of the study we start doing a test and evaluation.

Visual based vehicle detection system: In the real time traffic monitoring systems, data can come from a various sources like the ultrasonic detectors, loop detectors, radar

Corresponding Author: Mohammad Iqbal, Department of Computer Engineering,
Faculty of Computer Science and Information Technology, Gunadarma University, Depok, Indonesia

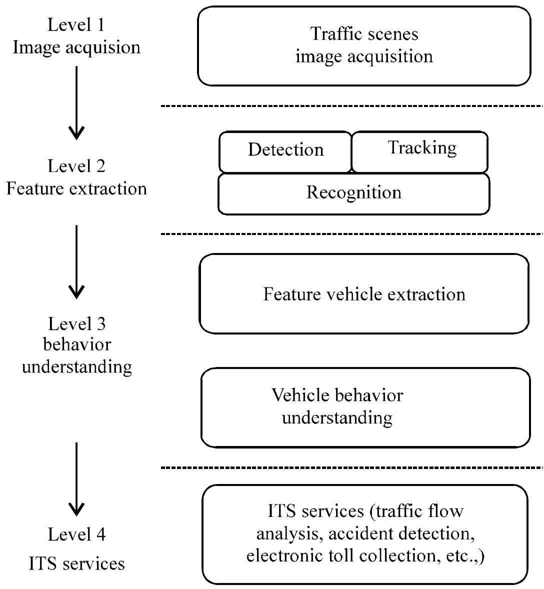

Fig. 1: Hierarchical vehicle surveillance

sensors, microwave sensors or video cameras. Due to the lastest acheivement in computer vision and image processing field, the video cameras have been found to be an efficient technique to collect and analyze the traffic data. Vehicle detection is the second step in video based analysis in ITS applications after image acquisition (Tian *et al*., 2015). Current methods of vehicle detection are already accurate and robust, therefor researches for the next step are influenced which is vehicle recognition. The research technique in this field was divided into; motion based and appearance based (Elkerdawi *et al*., 2014). Segmentation techniques based-on motion is using movement characteristic to separate moving vehicles from stationary background image. On the other hand, appearance based techniques is extracting features from appearance of the vehicle like color, texture and shape to distiguish the vehicle from the background (Fig. 1).

Previous research of vehicle detection step using appearance based explained the popular feature descriptors have been used to describe the vehicle's visual appearance is begin from the simple method based-on histogram but it sensitive to size the vehicle and the change of illumination. Then the improved method based-on histogram found, namely Histogram of Oriented Gradient (HOG) (Shakin and Vasuki, 2015). This method is based on the count of the occasions of gradient orientation of portions of image. HOG features sensitive to geometric and illumination invariance and efficient in computational SIFT (Narhe and Nagmode, 2014) however, has the advantages because of its invariance to scaling and rotation, it is also invariant to affine projection and changes of illumination. SIFT work by considering local edge orientations around stable keypoints. Another descriptor is Haar-like features which subtracts the sum of pixels from the rectangles over an image patch. Structure in rectangular of Haar-like features is founded very efficient to compute, making this method work properly for real-time applications to representing rigid moving objects such as vehicles in the road.

Hardware system: The hardware used in this prototype system consists of raspberry-Pi 3 model B board, a USB webcam and power supply that are arranged in stereo System.

Raspberry Pi is a small sized single board computer with a basic size of 85×56 mm (credit card size) developed by raspberry Pi foundation. It has various interfaces for interconnecting other devices, wide network connection support and also has a low power consumption (5 V, 2.5A). Micro SD port can be used to load operating system (linux based OS such as Raspbian, Fedora, Debian, etc.) and storing data. The basic software tool for computer vision application using in Raspberry Pi is Open CV with programming language like python, C/C++, Java, Ruby, etc. We use 5-megapixel webcam capable to capture 30 fps with 1024×768 resolution.

To capture images, Logitech usb webcam is utillized. The webcam has a 5 megapixel sensor resolution, 640×480 still image with 30 fps of frame rate and video capture resolution of 1024×768. This camera is easy to plug in and supported by Linux based OS.

Software requirement: We use openCV with python language. Open cv is a software library practically used for machine learning and computer vision. OpenCV is free for use for both academic and commercial use and is released under open source BSD licence.

Stereo vision for vehicle detection, classification and counting system: The system design is described in Fig. 2. The system is consists of two raspberry pi with camera that are arranged in stereo vision system both camera system doing vehicle detection using descriptors Haar Like features then the results are evaluated to find matching features between stereo images and if found, the system will do the vehicle counting. Then, the data is sent over a computer network to display in web application.

The left and right camera system will extract features using the Haar classifier cascade and make a box for each object that was detected successfully. The box will be processed further as a feature that will be compared between the left and right images to get a matching feature, if the box found in two images, the classification and counting of vehicle will be done based on the width and height of the box.

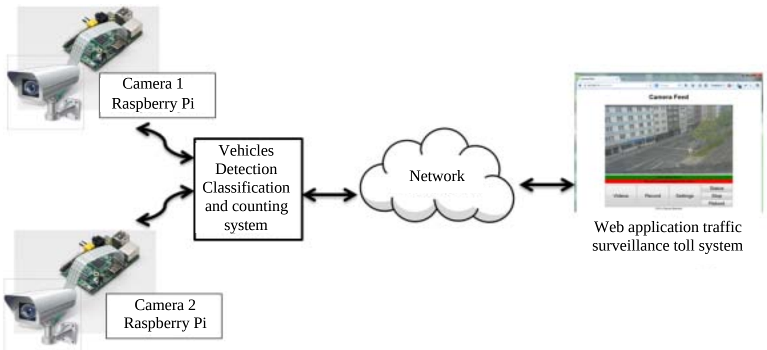

Fig. 2: Realtime stereo vision for vehicle detection, classification and counting system

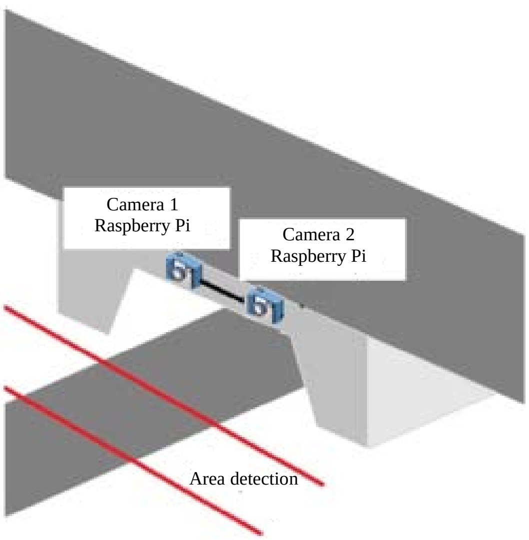

Fig. 3: Camera position in scenes

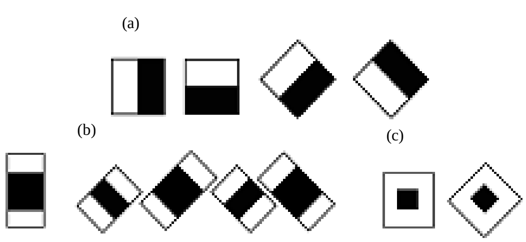

Fig. 4: Type of Haar-like features: a) edges; b) lines and c) center surrounded

The important thing to be considered in the stereo system is the distance between the two cameras are well known and area detection are defined in advance in this case we created a boundary using imaginary line on each image. This is done to ensure that the calculation of box to be more accurate. illustration camera position and area detection can be in Fig. 3.

Haar cascade classifier: Haar-like features in two dimensions are composed of adjacent light and dark rectangles. Figure 4 shows some typical features used for recognition. The approach to detect objects in the image combines four main concepts:

- Training data
- Extract features a simple rectangular called Haar features
- Integral image for rapid detection feature
- The classification in stages (cascade classifier) to connect a lot of features efficiently

First step we collect positive samples (image with object) and negative samples (image without object). The numbers of images sample are depended on a variety of factors such as the object type to be recognized, the quality of the images, the method to generate the samples and the processor power. Highly accurate classifier training need a huge number of samples and takes a lot of time.

Haar features: Once we obtained the data to be trained, the next step is to extract Haar features. Features are nothing other than region of window that are spread through the image and used as object detection classifier. Haar features have a rectangular shape (Fig. 4) and can be contained of several more rectangles in each of the feature. Each window will have either black or white region.

The value of two-rectangle feature can be calculated from subtraction of the sum of the pixels from two regions. On a three-rectangle feature, the value is calculated as a subtraction within two outside rectangles from the sum in a center rectangle. And in a four-rectangle feature we can do the subtraction diagonally within four regions.

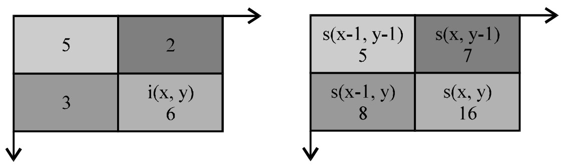

Fig. 5: Original image (left) and its integral image (right)

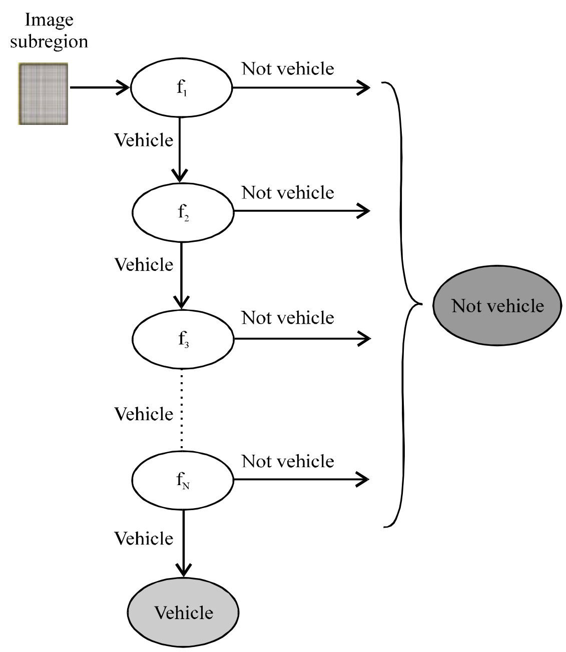

Fig. 6: The classifier cascade as a chain of filters

Integral image: The subtraction of the sum of the pixels approach can be exhaustive if we apply this method directly through the image considering that there are so many regions to be calculated from the image. And seeing that we are implementing this method on a sequential real time data, we need our algorithm to be fast.

To accelerate this process, we transform the image to integral image (Messom and Barczak, 2006) which is a table of the sum of pixels to the left and above of (x, y). The integral image is calculated by:

$$s(x, y) = i(x, y)+s(x-1, y)+s(x, y-1)-s(x-1, y-1)$$

For better representation of how we calculate the integral image can be seen in following example illustrated in Fig. 5. Once we obtained the integral image we can easily calculate the sum of any region quickly. To do this, we use this equation:

$$i(x', y') = s(A)+s(D)-s(B)-s(C)$$

Stages of classification: Haar features is weak to detect object on an image by itself, therefore we have to implement many features set and chain them to obtain accurate object detection. To do this we use Adaboost. Adaboost combine many classifier into one classifier. Each classifier has a threshold. From this combination of classifier we can have a really strong and accurate classifier.

The filter on each level is trained to classify the images from the training set. If one from the filters fail to detect the object from a region, then that region will be classified as "not vehicle", then the region window will continue to move. When one filter succeeded to detect the object then that region will passed to the next filter. Image region that has passed through all the filters will be classified as "vehicle". Illustration can be seen at Fig. 6.

Stereo vision system: The stereo vision has two main problems, first is correspondence problem which is how to find an association point in the left image from the right image. Second is reconstruction problem which is how to construct 3D points from matched points.

Reconstruction problem has been done by triangulation. The existence of noise make the correspondence problem more challenging. The noise can be distortions in geometric and photometric, repetitive patterns, occlusions and discontinuities. Epipolar geometry is used here by implementing epipolar line to search correspondence point. It works by finding point on conjugated epipolar lines which is a straight line that conjugate the epipolar plane, a plane from the object, reference image and target image (Fig. 7).

In our case, we use local methods matching NSSD (Normalized-Sum of Squared Differences) (Narhe and Nagmode, 2014) because observations were made on a pixel level from Haar like feature. Correspondence point is discovered by finding similarity of predefined neighborhood or window. Two inputs from left image and right image are compared by local matching.

Realtime stereo vision for vehicle detection algorithm: The algorithm composed in this study is:

- Extract features using the Haar classifier cascade for stereo camera system
- Apply ratio test to find best matches (NSSD), the locations of matched keypoints are then extracted from both images to find the perspective transformation
- Some possible errors while matching can be verified by RANSAC algorithm. This algorithm will provide information of good matches (inliers) and bad matches (outliers)

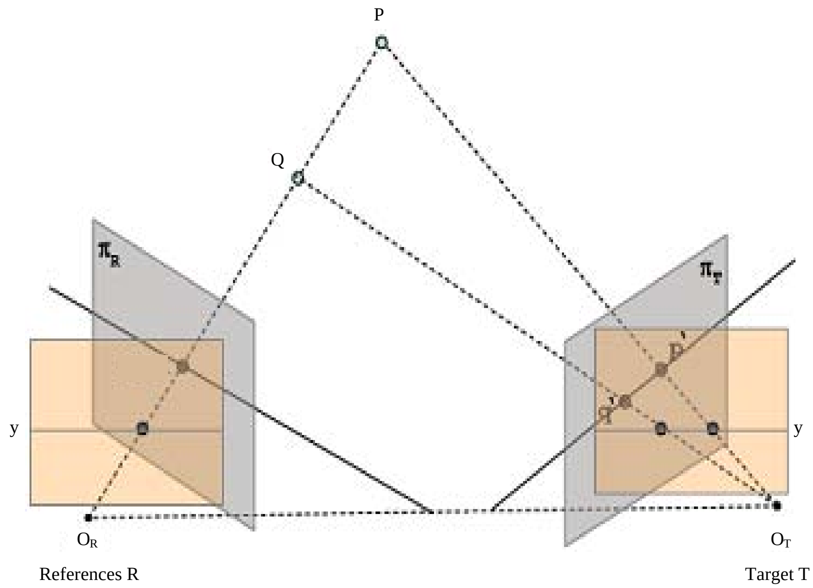

Fig. 7: Epipolar geometry as corespondence problem solution in stereo vision

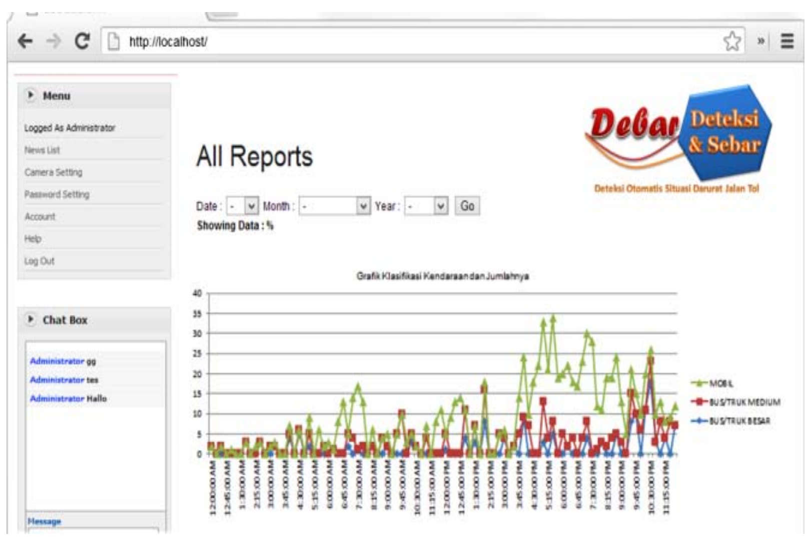

Fig. 8: Display report current day monitoring in web application

Displaying the result in web application: In communication process, the vehicle classification and counting data result will send to database in web server. To display infographics in a website can use PHP scripts from JPGraph that can create charts from the database (Fig. 8).

RESULTS AND DISCUSSION

Evaluation is done by several test scenarios to get the desired results. The tested scenes includes time, weather and traffic conditions. The period of time in this research is in the morning, noon and night. While weather conditions were chosen is sunny and cloudy. To maximize scenario of testing we used the traffic conditions at no congestion, half congested and traffic jams. All of these conditions will also be combined with each other. The description of test scenes can be seen in Fig. 9.

Evaluation of the system is done by direct observation using two systems simultanously. We compute success rate by running the vehicle counting based-on Haar-like features in two cameras and calculate

Table 1: Average result in left/right camera (counting based-on using haar-like features)

Traffic condition	Average result in left/right Total vehicle	Vehicle detected	Wrong detection	Success rate (%)
Morning-clear-no-congestion	10	10	0	100.0
Morning-clear-half-congestion	16	15	1	93.3
Morning-clear-traffic-jam	39	18	21	46.1
Morning-cloudy-no-congestion	12	12	0	100.0
Morning-cloudy-half-congestion	18	16	2	89.0
Morning-cloudy-traffic-jam	30	15	15	50.0
Afternoon-clear-no-congestion	10	10	0	100.0
Afternoon-clear-half-congestion	16	16	0	100.0
Afternoon-clear-traffic-jam	35	20	15	57.2
Afternoon-cloudy-no-congest ion	12	12	0	100.0
Afternoon-cloudy-half-congestion	15	15	0	100.0
Afternoon-cloudy-traffic-jam	38	18	20	47.4
Night-clear-no-congestion	24	24	0	100.0
Night-clear-half-congestion	18	18	0	100.0
Night-clear-traffic-jam	35	15	20	43.0
Night-cloudy-no-congestion	8	8	0	100.0
Night-cloudy-half-congestion	15	14	1	93.3
Night-cloudy-traffic-jam	30	11	19	37.0

Table 2: Counting result based-on stereo matching two images in left and right camera)

Traffic condition	Counting result based-on stereo Total vehicle	Vehicle detected	Wrong detection	Success rate (%)
Morning-clear-no-congestion	10	10	0	100.0
Morning-clear-half-congestion	16	16	0	100.0
Morning-clear-traffic-jam	39	33	6	84.6
Morning-cloudy-no-congestion	12	12	0	100.0
Morning-cloudy-half-congestion	18	17	1	94.0
Morning-cloudy-traffic-jam	30	25	5	83.0
Afternoon-clear-no-congestion	10	10	0	100.0
Afternoon-clear-half-congestion	16	16	0	100.0
Afternoon-clear-traffic-jam	35	30	5	86.0
Afternoon-cloudy-no-congest ion	12	12	0	100.0
Afternoon-cloudy-half-congestion	15	15	0	100.0
Afternoon-cloudy-traffic-jam	38	31	7	82.0
Night-clear-no-congestion	24	24	0	100.0
Night-clear-half-congestion	18	18	0	100.0
Night-clear-traffic-jam	35	33	2	94.0
Night-cloudy-no-congestion	8	8	0	100.0
Night-cloudy-half-congestion	15	15	0	93.3
Night-cloudy-traffic-jam	30	29	1	97.0

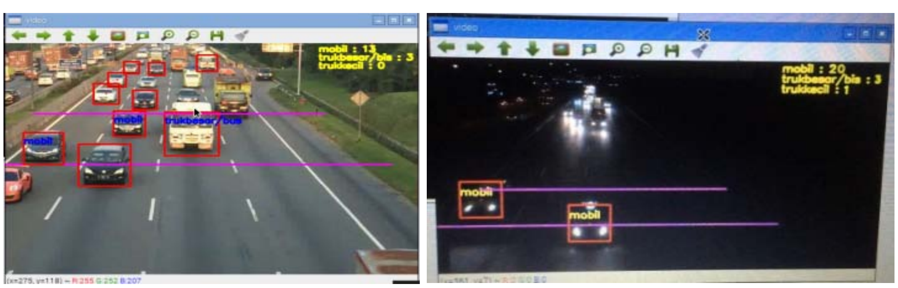

Fig. 9: Scenario testing in morning-clear-half congested (left) and night-clear-no congestion (right)

the average, then it compared with the results obtained from the vehicle counting using a stereo matching. The average calculation of vehicle types using Haar-like features can be seen in Table 1.

The results of vehicle type counting based stereo matching can be in Table 2. We can see improvement detection capability of the stereo system, especially on test in traffic jam scenes. This occurs due to the blocked vehicle by other vehicles, still can be detected by the two points of view of the stereo vision.

CONCLUSION

This study proposed a method using Haar-like feature and the stereo vision system in order to detect vehicles. Haar-like feature is used to obtain stronger feature for realtime monitoring. Stereo vision is then implemented to obtain more reliable detection result by using matching system from two cameras to verify vehicle candidate in real time. The execution time of the proposed algorithm litle bit longer than only using Haar-like features in one camera but still proper to apply it to the real-time system. It has limitation that error sometimes occur when the illumination of the road environment changes.

ACKNOWLEDGEMENTS

This researche was supported by Ministry of Research, Technology and Higher Education of the Republic of Indonesia in Research schema "Penelitian Unggulan Perguruan Tinggi". I would say thank you very much for all researchers and student in Gunadarma University for their hard work in this research, specially to Rachmat Sampurna, Rendi Nurcahyo.

REFERENCES

Elkerdawi, S.M., R. Sayed and M. ElHelw, 2014. Real-Time Vehicle Detection and Tracking using Haar-Like Features and Compressive Tracking. In: ROBOT2013: First Iberian Robotics Conference, Armada, M.A., A. Sanfeliu and M. Ferre (Eds.). Springer, Berlin, Germany, pp: 381-390.

Messom, C. and A. Barczak, 2006. Fast and efficient rotated Haar-like features using rotated integral images. Proc. Aust. Rob. Autom., 1: 1-6.

Narhe M.C. and M.S. Nagmode, 2014. Vehicle classification using SIFT. Int. J. Eng. Res. Technol. (IJERT.), 3: 1735-1738.

Shakin, A.B. and P. Vasuki, 2015. Video based vehicle detection using morphological operation and HOG feature extraction. ARPN. J. Eng. Appl. Sci., 10: 1866-1871.

Tian, B., B.T. Morris, M. Tang, Y. Liu and Y. Yao *et al*., 2015. Hierarchical and networked vehicle surveillance in ITS: A survey. IEEE. Trans. Intelligen. Transp. Syst., 16: 557-580.