Adaptive Vehicle Detector Approach for Complex Environments

M.P. Sirisha
Assistant Professor
Department of Electronics and Communication Engineering
Vardhaman College of Engineering , Hyderabad, Andhra Pradesh , India

A.Shanthi
Department of Electronics and Communication Engineering
Vardhaman College of Engineering, Hyderabad, Andhra Pradesh , India

Abstract- Video based vehicle detection technology is an integral part of Intelligent Transportation System (ITS), due to its non-intrusiveness and comprehensive vehicle behavior data collection capabilities Vehicle detection under various environments will have many difficulties such as illumination vibrations, shadow effects and vehicle overlapping problems that appear in traffic jams. This paper presents a Vision based Vehicle counting and vehicle speed calculation. The system detects the vehicles, counts them and calculates speed of each vehicle. Several computer-vision based algorithms were developed or applied to extract foreground objects from a video sequence, detect presence of vehicles detection, counting .The algorithms were implemented in the system using C++. The system uses the Intel OpenCV library for image processing. This paper is implemented using ARM9 micro controller. The webcam is connected to the controller through USB device. The controller processes the information and monitors the results as vehicles and number of vehicles on remote controlled PC through Ethernet and also the information is displayed on LCD unit.

Keywords – Image processing, Background Extraction, Foreground Extraction Vehicle Detection, Vehicle Counting, ITS , ARM9

I. INTRODUCTION

Monitoring traffic using a vision oriented system in crowded traffic areas is a challenging task. Current approaches of monitoring traffic include manual counting of vehicles, or counting vehicles using magnetic loops on the road. The main drawback of these approaches, besides the fact that they are expensive, is that these systems only count. Vehicle speed and location registration for individual vehicles are properties that are not, or only limited, supported by these systems. In this paper an economical and versatile traffic monitor system is proposed using a vision based approach. In this system, traffic at intersections in cities is monitored using simple cameras that are located on a high spot somewhere near these intersections. A typical video frame obtained from such cameras is displayed. Most existing research and applications on traffic monitoring focus on monitoring cars on highways. As will become clear in next sections; Many different vision-based vehicle detection systems have been developed. These studies can be categorized into three classes: (i) vehicle detection, (ii) vehicle verification, and (iii) vehicle tracking. All the studies are related to our study and surveyed here. Video-based traffic monitoring systems [1] have number of advantages over traditional methods, such as loop detectors. In addition to vehicle counting, more traffic information can be obtained by video images. Real-time vehicle detection in a video stream relies heavily on image processing techniques, such as motion segmentation, edge detection and digital filtering, etc. different vision-based vehicle detection approaches have been proposed by researchers [2-5] Video-based real-time vehicle detection method by classed background learning[6] is a system developed background subtraction is used for vehicle detection method The key technique of video-based vehicle detection belongs to a classic problem of motion segmentation. One of the widely used techniques to identify moving objects (vehicles) is background subtraction or background learning [7]. Furthermore, recent studies [8, 9, 10] evaluating some of these commercial systems found that shadows and headlight rejections generated significant problems of false positives and early detections. The paper studies the detection of oncoming vehicles in track scenes by using depth information. The image sequences in our experiments are captured by a pair of stereo cameras which are mounted in a test vehicle. This work focuses on the recognition of oncoming vehicles. Although stereo vision by itself is not reliable enough to perform accurate vehicle detection it is useful to quickly generate object hypotheses which can then be varied by accurate pattern recognition techniques. V. Kamat, Altan, S. Ganesan. This paper describes a vision based vehicle identification algorithm. This algorithm has been developed using image processing and pattern recognition techniques. This paper describes the role of the image processing techniques in this algorithm, specially, the application of the Hough Transform in this scheme[11]. This paper is motivated to develop a new video-based vehicle detection system for convenient, reliable, economic traffic data collection using images captured by video cameras then using computer vision algorithm to extract the vehicle from the video then detect the vehicle, then counting and extracting some other parameters like motion parameters, and tracking of multiple vehicles , speed.

The proposed system will use the OpenCV library from Intel for image processing. The video shall be read frame by frame, and motion detection algorithms used to detect the image region of the moving vehicle. Thus each moving object identified as vehicle (size, speed etc..) is used for counting the number of vehicles. This system is implemented in C++ with OpenCV and wxWidgets Library. This approach use a camera to capture video of vehicle pass through highways. The information is displayed on ARM board and also on PC through Ethernet.

The rest of the paper is organized as follows. System Overview is presented in section II . Proposed algorithms are explained in section III. Experimental results and conclusions are presented in section IV.

II . SYSTEM OVERVIEW

This proposed System deals with detecting the vehicles, counting the vehicles on highways. First we should capture the video of the moving vehicles from appropriate location. This video information is used as the input for the entire project. The video is extracted to successive frames (images) and output image is a color image. And background extraction is applied to the moving images and foreground and background objects are separated. Then various counting and algorithm is applied on the parameters extracted from the. Intel OpenCV library is used for image processing and C++ programming language is used to develop the system. The main intention of this system is to detect the vehicle in complex environments. To detect the vehicle we will use camera which is interfaced to a micro controller. Using camera we will continuously capture or continuously record the video stream like vehicles, no. of vehicles in the area, computes the information and transfers the compressed video stream to the ARM micro controller. This project is based on Linux operating system (ubuntu Linux)

ARM9 is an ARM architecture 32-bit RISC CPU family. With this design generation, ARM moved from a von Neumann architecture (Princeton architecture) to a Harvard architecture with separate instruction and data busses (and caches), significantly increasing its potential speed. Most silicon chips integrating these cores will package them as modified Harvard architecture chips, combining the two address busses on the other side of separated CPU caches and tightly coupled memories.

S3C2440A Microcontroller:
SAMSUNG's S3C2440A 16/32-bit RISC microprocessor. SAMSUNG’s S3C2440A is designed to provide hand-held devices and general applications with low-power, and high-performance microcontroller solution in small die size. To reduce total system cost, the S3C2440A includes the following components. The S3C2440A is developed for cost- and power-sensitive applications. It adopts a new bus architecture known as Advanced Micro controller Bus Architecture (AMBA). The S3C2440A offers outstanding features with its CPU core, a 16/32-bit ARM920T RISC processor designed by Advanced RISC Machines, Ltd. The ARM920T implements MMU, AMBA BUS, and Harvard cache architecture with separate 16KB instruction and 16KB data caches, each with an 8-word line length. By providing a complete set of common system peripherals, the S3C2440A minimizes overall system costs and eliminates the need to configure additional components.

Linux refers to the family of Unix-like computer operating systems using the Linux kernel. Linux can be installed on a wide variety of computer hardware, ranging from mobile phones, tablet computers and video game consoles, to mainframes and supercomputers. Linux is the leading server OS, accounting for more than 50% of installations. Desktop use of Linux has increased in recent years, partly owing to the popular Ubuntu, Fedora, and open USE distributions and the emergence of net books and smart phones running an embedded Linux. The name "Linux" comes from the Linux kernel, originally written in 1991 by Linus Torvalds. The main supporting user space system tools and libraries from the GNU Project (announced in 1983 by Richard Stallman) are the basis for the Free Software Foundation's preferred name *GNU/Linux*.

OpenCV is an open source computer vision library. The library is written in C and C++ and runs under Linux, Windows and Mac OS X. OpenCV was designed for computational efficiency and with a strong focus on real- time applications. Computer vision is a rapidly growing field, partly as a result of both cheaper and more capable cameras, partly because of affordable processing power, and partly because vision algorithms are starting to mature. OpenCV itself has played a role in the growth of computer vision by enabling thousands of people to do more productive work in vision. With its focus on real-time vision, OpenCV helps students and professionals efficiently implement projects and jump-start research by providing them with a computer vision and machine learning infrastructure that was previously available only in a few mature research labs.

Qt is a cross-platform application framework that is widely used for developing application software with a graphical user interface (GUI) (in which cases Qt is classified as a *widget toolkit*), and also used for developing non-GUI programs such as command-line tools and consoles for servers. Qt uses standard C++ but makes extensive use of a special code generator (called the *Meta Object Compiler*, or *moc*) together with several macros to enrich the language. Qt can also be used in several other programming languages via language bindings. It runs on the major desktop platforms and some of the mobile platforms. It has extensive internationalization support. Non-GUI features include SQL database access, XML parsing, thread management, network support, and a unified cross-platform application programming interface (API) for file handling.

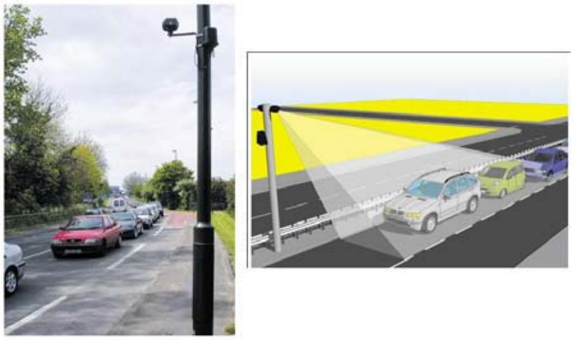


Fig 1: Mounted camera on lamp post to capture vehicle video

III. PROPOSED ALGORITHM

The proposed system uses linux operating system, opencv library ,wxwidegets. The video input is given to ARM processor. A live video capture module is developed to digitize live video signals into image frames from common video sources, such as a surveillance cameras or Digital camera. Camera is mounted on a lamp post in such a way that the occlusions of vehicle moving toward both side is less .so that we get a top long view of vehicles as shown in fig 1 , the image format of the Joint Photographic Experts Group (JPEG) and the video format AVI (Audio Video Interleave).

A: Filtering

OpenCV provides a fast, convenient interface for doing *morphological transformations* on an image. The basic morphological transformations are called *dilation* and *erosion*, and they arise in a wide variety of contexts such as removing noise, isolating individual elements, and joining disparate elements in an image. Morphology can also be used to find intensity bumps or holes in an image and to find image gradients.

Dilate :

$dist(x,y)= \max_{(x,x'):element(x,y')\neq 0} src(x+x',y+y')$

Erode :

$dist(x,y)= \min_{(x,x'):element(x,y')\neq 0} src(x+x',y+y')$

Dilating: This operation consists of convoluting an image A with some kernel B which can have any shape or size, usually a square or circle. The kernel B has a defined *anchor point*, usually being the center of the kernel. As the kernel B is scanned over the image, we compute the maximal pixel value overlapped by Band replace the image pixel in the anchor point position with that maximal value. As you can deduce, this maximizing operation causes bright regions within an image to "grow" (therefore the name *dilation*). Shown in fig 3.(a).

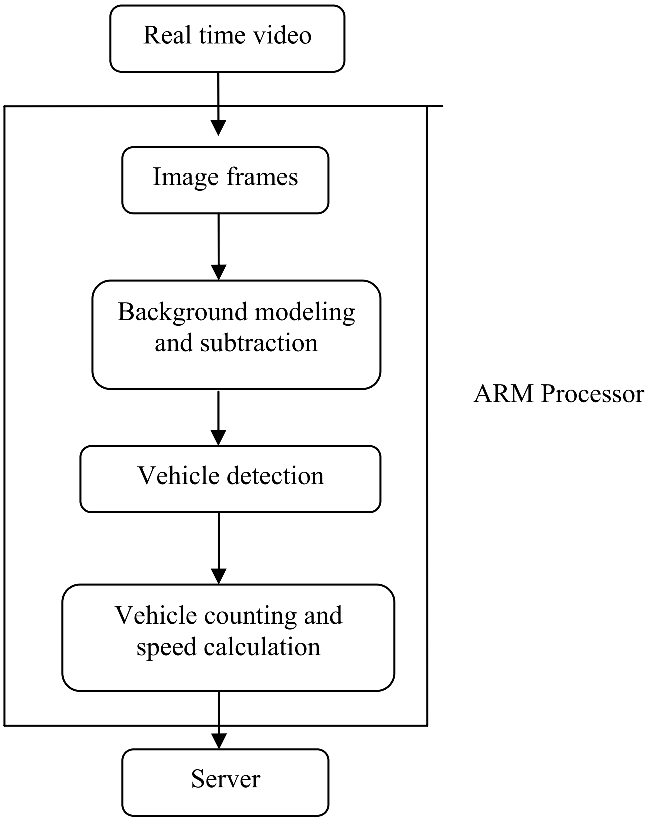

Real time video
Image frames
Background modeling and subtraction
Vehicle detection
Vehicle counting and speed calculation
Server
ARM Processor

Fig 2: System overview of proposed system

Eroding: This operation is the sister of dilation. What this does is to compute a local minimum over the area of the kernel. As the kernel B is scanned over the image, we compute the minimal pixel value overlapped by B and replace the image pixel under the anchor point with that minimal value. Analagously to the example for dilation, we can apply the erosion operator to the original image (shown above). You can see in the result below that the bright areas of the image (the background, apparently), get thinner, whereas the dark zones (the "writing"(gets bigger. Shown in fig 3(b)

B: Background Extraction: In the scene, what interests us is the region corresponding to the motion (moving vehicles) rather than the entire view. Hence, the next step of the system is to isolate this region (called foreground) from the rest of the frame (called background).

Foreground:
The spatially small region of the frame which changes in a relatively short time interval (e.g. people/ Vehicle / animals). As shown in fig 4 (a).

Background:
The spatially large region of the frame which changes very slightly if at all during a long enough time interval (e.g. road / trees / buildings). Foreground extraction is used for separating the moving objects (vehicle) from the source video, only if we can successfully segment foreground can the following object detection, identification and tracking procedures be successful. So segmentation is the foundation to the whole surveillance system. As shown in fig 4 (b).

C: Vehicle Detection Module

Vehicle Detection is the process of identifying the vehicle that we extracted from the background, for identifying each vehicle we need to consider each vehicle by drawing rectangles to detect and count vehicles. The controller draws two regions. The no. of rectangles present at first region shows no. of vehicles present in particular region. And also the controller will track the vehicles and calculate in how many frames the rectangle will move from first region to second region gives speed of the vehicle.

D: Working Principle

The main intention of this system is to detect the vehicle in complex environments. To detect the vehicle we will use camera which is interfaced to a micro controller. Using camera we will continuously capture or continuously record

the video stream like vehicles, no. of vehicles in the area, computes the information and transfers the compressed video stream to the ARM micro controller. The proposed motion detection system makes use embedded board which makes use of less power consumptive and advanced micro controller like S3C2440. S3C2440 is a Samsung company's microcontroller which is designed based on the structure of ARM 920T family. This microcontroller works for an voltage of +3.3V DC and at an operating frequency of 400 MHz, The maximum frequency up to which this micro controller can work is 533 MHz We cannot get S3C2440 microcontroller individually. We will get it in the form of FRIENDLY ARM board otherwise we can call it as MINI 2440 board. In order to work with ARM 9 micro controllers we require 3 things. They are as follows.1. Boot Loader2. Kernel 3. Root File System
Boot loader: The main functionality of boot loader is to initialize all the devices that are present on the mother board of MINI 2440 and at the same time to find out whether any problem or any other fault is there in the devices that are present on that mother board of MINI 2440. The other feature of the boot loader is to find out what are the different operating systems that are present in the standard storage devices and to show it on to the display device so that user can select between the operating systems into which he wants to enter. One other feature of the boot loader is to load operating system related files byte by byte into the temporary memory like RAM. In our current project we are using boot loader like Super vivi which is MINI 2440 specific.

Kernel: The core part of an operating system we can cal like kernel. Operating system will perform its functionalities like File management, Process management, Memory management, Network management and Interrupt management with the help of the kernel only. Kernel holds the device related drivers that are present on the motherboard. Friendly arm board supports for operating systems like symbion, android, embedded Linux, win CE but in all these operating systems embedded Linux will provide high security to drivers and files. So in our current project we are making use of kernel of embedded Linux with which device related drivers that are present on the mother board of friendly arm board will automatically come when we load embedded Linux related kernel.
Root File System: File system will tell how the files are arranged in the internal standard storage devices. In embedded Linux, kernel treats everything as a file even the input and output devices also. In embedded Linux, Root is the parent directory it contains other sub directories like dev, lib, home, bin ,sbin ,media ,mnt ,temp ,proc , etc, opt and etc. According to our application we will interface some external devices also. All the devices means internal devices that are present on the motherboard of MINI 2440 will get their corresponding drivers when we load Embedded Linux related kernel. But these device drivers require micro controller related header files and some other header files which will be present in the lib directory which is present in the root directory. And also the devices related drivers will be present in the dev directory which is again present in the root directory. So whenever we will load the Root File System then we will get different directories which will be helpful to the kernel. So compulsorily we need to load the Root File System. MINI 2440 specific Root File System is Root Qtopia.
The essential programs that are required in order to work with MINI 2440 like Boot loader, Embedded Linux related Kernel, Root File System will be loaded into the NOR flash which is present on the MINI 2440 board itself. The program that is related with the application will be loaded into NAND flash which is also present on the MINI 2440 board itself. By using boot strap switch that is present on the MINI 2440 will help the user to select either NOR or NAND flash. After that by using DNW tool we can load Boot loader, Embedded Linux related kernel and Root File System into NOR flash by using USB cable and the application related program into NAND flash. Once loading everything into MINI 2440 board it starts working based on the application program that we have loaded into the NAND flash.

By using USB type camera that is interfaced to the embedded board we can capture the live video of vehicle movement from the particular location. To detect the motion first Open the video device, and capture the video from video device and grab the frame from video. Then read the image from the grab frame and store that image in particular memory location. After that we will read already stored image, then the current image and already stored image will convert into gray image. Then compare both the images and differentiate them. Again we will convert that image into black and white image. And draw contour to that image. If any difference will get draw a rectangle. The controller draws two regions. The no. of rectangles present at first region shows no. of vehicles present in particular region. And also the controller will track the vehicles and calculate in how many frames the rectangle will move from first region to second region gives speed of the vehicle. In this way the controller will detect no. of vehicles and calculate speed of each vehicle in a particular region. As shown in fig 5. And also the controller will send the information in the form of frame by frame by using FTP protocol to the remote PC through Ethernet technology.

III. EXPERIMENT AND RESULT

The proposed motion detection system makes use embedded board which makes use of less power consumptive and advanced micro controller like S3C2440. S3C2440 is a Samsung company's microcontroller which is designed based on the structure of ARM 920T family. This microcontroller works for an voltage of +3.3V DC and at an operating frequency of 400 MHz, The maximum frequency up to which this micro controller can work is 533 MHz We cannot get S3C2440 microcontroller individually. We will get it in the form of FRIENDLY ARM board otherwise we can call it as MINI 2440 board.

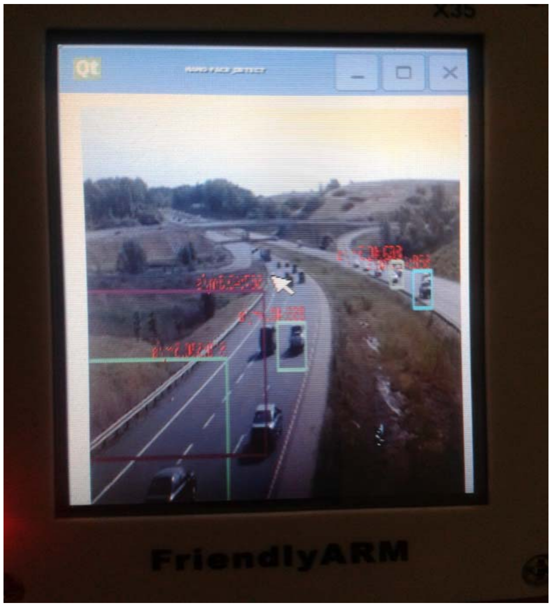
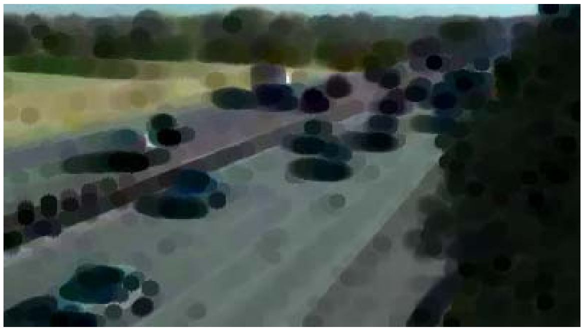


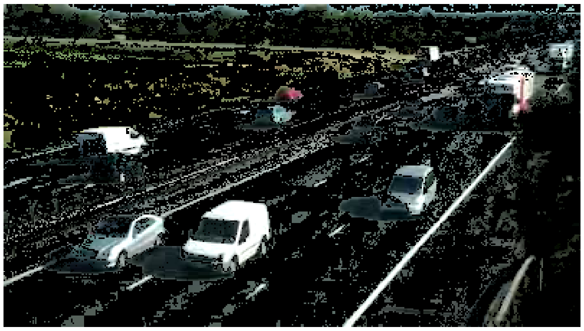
Fig: 3 ARM9 S3C2400

(a)

(b)

(b)

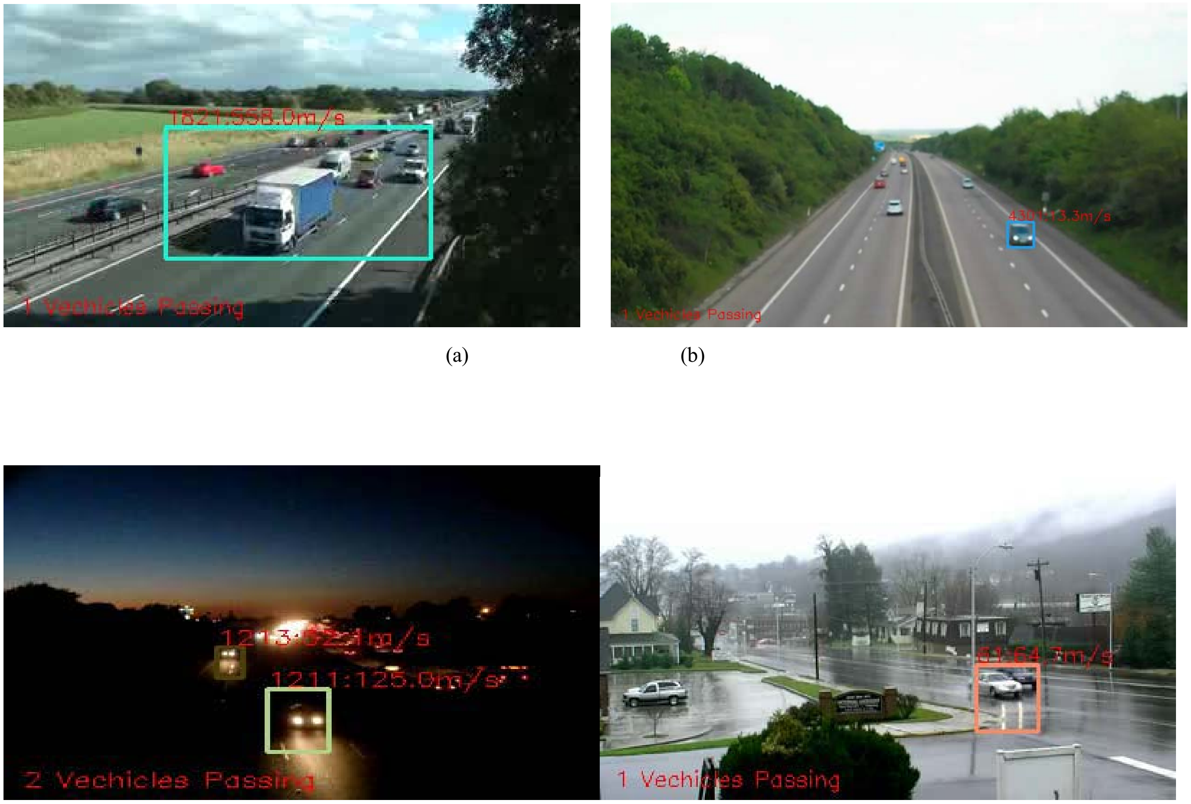




(c)

Fig 4: (a) Source image (b) Dilation (c) erosion

Fig 5: (a) source image (b) background (c) foreground



(a) (b)

© (d)

Fig :6 Experiment scenarious for different weather conditions: (a) Sunny (b) Evening (c)Night(d) Rainy

IV.CONCLUSION

The project **“Adaptive Vehicle Detection Approach for Complex Environments”** has been successfully designed and tested. It has been developed by integrating features of all the hardware components and software used. Presence of every module has been reasoned out and placed carefully thus contributing to the best working of the unit. Secondly, using highly advanced ARM9 board and with the help of growing technology the project has been successfully implemented.

V. FUTURE WORK

The computer vision and image processing can be applied in many aspects for the traffic parameter extraction (Vehicle counting, speed measurement etc). So far, we succeeded to get the number of vehicle, vehicle speed etc, the accuracy of the result can be further improved by some good background extraction, now vehicle occlusion and shadows are problem for getting the correct value. Currently the application is working in a standalone system, if we implement this application to multiple locations in the town using web application then the data can be further used for traffic controlling and planning. The vehicle detection in the proposed system is robust in variant weather conditions.

REFERENCES

[1] A video-based real-time vehicle detection method by classed background learning Xiao-jun Tan, Jun Li Chunlu Liu Sun Yat-sen University, Guangzhou,People's Republic of China,Deakin University, Geelong, Australia.

[2] Kim, J.B. et al, Wavelet-based vehicle tracking for automatic trac surveillance. Proc. IEEE Region 10 Inter. Conf. on Electrical and Electronic Technology , Singapore, 1, 313-316 (2001).

[3] Lee, J.W. et al, A study on recognition of lane and movement of vehicles for port AGV vision system. Proc. 2002 IEEE Inter. Symp. on Industrial Electronics , L'Aquila, Italy, 2, 463-466 (2002).

[4] Chen, S.C. et al, Spatiotemporal vehicle tracking the use of unsupervised learning-based segmentation and object tracking spatiotemporal vehicle tracking. Robotics & Automation Mag ., 12, 1 , 50-58 (2005).

[5] Wang, Y.K. and S.H. Chen. Robust vehicle detection approach. Proc.IEEE Conf. on Advanced Video and Signal Based Surveillance , Como, Italy,117-122 (2005).

[6] Xie, L. et al, Real-time vehicles tracking based on Kalman lter in a videobased ITS. Proc. 2005 Inter. Conf. On Communications, Circuits & Systems , Hong Kong, China, 2, 883-886 (2005).

[7] Friedman, N. and Russell, S., Image segmentation in video sequences: a probabilistic approach. Proc. 13th Conf. On Uncertainty in Arti_cial Intelligence, Providence, USA (1997).

[8] Bonneson, J. and M. Abbas. Video Detection for Intersection and Inter change Control. FHWA/TX-03/4285-1. Texas Transportation Institute. College Station, Texas, 2002.

[9] Martin, P.T., G. Dharmavaram, and A. Stevanovic. Evaluation of UDOT's Video Detection Systems: System's Performance in Various Test Conditions. Report No: UT-04.14. Salt Lake City, Utah, 2004.

[10] Rhodes, A., D.M. Bullock, J. Sturdevant, Z. Clark, and D.G. Candey,Jr. Evaluation of Stop Bar Video Detection Accuracy at Signalized Intersec-tions. CD-ROM. Transportation Research Board of the National Academies,Washington D.C., 2005.

[11] HOUGH TRANSFORM FOR VEHICLE IDENTIFICATION V. Kamat, 0. Altan, S. Ganesan, Department of Computer Science and Engineering, Oakland University Rochester, M148309-4401

[12] Adaptive Vehicle Detector Approach for Complex Environments Bing-Fei Wu, *Fellow, IEEE*, and Jhy-Hong Juang IEEE transactions on intelligent transportation systems, vol. 13, no. 2, june 2012

[13] Learning OpenCV by *Gary Bradski and Adrian Kaehler.*

[14] http://opensource.org

[15] http://SourceForge.net/projects/opencvlibrary