VEHICLE TRACKING AND SPEED ESTIMATION FOR TRAFFIC SURVEILLANCE

KAIRIL FARIQ BIN CHAIROL MOHD FEROZ

This project report is submitted in partial fulfillment of the requirement for the award of the Degree of Master of Electrical Engineering

Faculty of Electrical and Electronic Engineering
Universiti Tun Hussein Onn Malaysia

JULY, 2014

For my beloved father and mother

ACKNOWLEDGEMENT

It is with great reverence that I wish to express my deepest gratitude towards my supervisor, **Dr. Babul Salam B. Ksm Kader Ibrahim** for his untiring support, understanding, and valuable knowledge that helped immensely in making this project a success. His guidance and encouragement always kept my spirits up during work. I would also like to thank my **family** for being the best family a person could ask for; this is for them. My experience in working on this project has been great. My hope is that everything that I've learned and gained through this project will help me in future endeavours.

ABSTRACT

Vehicle tracking is one of the critical applications of object detection and tracking. Traffic surveillance has become crucial in this day and age where the number of vehicles on the road has risen considerably. To preserve the safety of motorists, traffic law enforcement assign speed limits at different locations throughout the country. However, irresponsible motorists still exceed the speed limit since they know it is unlikely that they will get caught. In this paper, a system is developed which is capable of detecting moving vehicles in a video and display the vehicles speed as it goes. Should a vehicle exceed the allowed speed limit, it will be displayed in the video alongside the vehicle so that traffic law enforcers will be able to take necessary action based on the displayed speed. The system uses Matlab/Simulink as a simulation platform as it provides comprehensive tools for thresholding, filtering and blob analysis. Optical flow was the image processing technique used to determine the moving vehicles. A median filter was used to remove salt and pepper noise from the thresholded image. Combinations of several morphological operations were used to rectify whatever that is left. Blob analysis produces rectangles around the moving objects. The centroid of the rectangle is used to determine the location of each vehicle at a given frame. To make up for the absence of depth perception, the camera's height and angle from the road is fixed so that the rate of which a vehicle approaches the camera can be determined. The results show that the system successfully detects vehicles and displays its speed, though there is a relatively small margin of error for the displayed speed. The displayed speed is set to only change once every couple of frames so that it would be easier to see.

ABSTRAK

Mengesan kenderaan adalah salah satu aplikasi kritikal pengesanan objek dan pengesanan. Pengawasan trafik telah menjadi penting dalam zaman ini di mana bilangan kenderaan di jalan raya telah meningkat dengan ketara. Untuk memelihara keselamatan pengguna jalan raya, penguatkuasaan undang-undang lalu lintas memberikan had laju di lokasi yang berbeza di seluruh negara. Walau bagaimanapun, pengguna jalan raya yang tidak bertanggungjawab masih melebihi had laju kerana mereka tahu bahawa kebarangkalian mereka akan ditangkap amat rendah. Dalam kertas ini, sistem yang dibangunkan mampu mengesan kenderaan yang bergerak dalam video dan memaparkan kelajuan kenderaan tersebut. Sekiranya kenderaan melebihi had laju yang dibenarkan, ia akan dipaparkan dalam video bersama kenderaan supaya penguatkuasa undang-undang trafik akan dapat mengambil tindakan yang sewajarnya. Sistem ini menggunakan Matlab / Simulink sebagai platform simulasi kerana ia menyediakan alat yang menyeluruh untuk *thresholding*, *filtering* dan *blob analysis*. *Optical flow* adalah teknik pemprosesan imej yang digunakan untuk menentukan kenderaan yang bergerak. *Median filter* telah digunakan untuk menyingkirkan *salt and pepper noise* daripada imej. Gabungan beberapa *morphological operations* telah digunakan untuk membetulkan apa yang kiri. *Blob analysis* menghasilkan kotak segi empat panjang di sekeliling objek yang bergerak. *Centroid* kotak segi empat panjang digunakan untuk menentukan lokasi setiap kenderaan di *frame*. Untuk membuat untuk ketiadaan *depth perception*, ketinggian kamera dan sudut dari jalan ditetapkan supaya kadar di mana kenderaan menghampiri kamera boleh ditentukan. Keputusan menunjukkan bahawa sistem itu berjaya mengesan kenderaan dan memaparkan kelajuan, walaupun ada margin kesilapan yang agak kecil untuk kelajuan yang dipaparkan. Laju yang dipaparkan hanya menukar sekali setiap beberapa *frame* supaya ia akan menjadi lebih mudah untuk dilihat.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

UAV	Unmanned Aerial Vehicle
SIFT	Scale-Invariant Feature Transform
SURF	Speeded Up Robust Features
3D	Three Dimensional
PNSR	Peak Signal to Noise Ratio
RGB	Red, Green, and Blue
2D	Two Dimensional
FPS	Frame Per Second
GPS	Global Positioning System
VHDL	VHSIC Hardware Description Language
HDL	Hardware Description Language
FPGA	Field-Programmable Gate Array
ASIC	Application Specific Integrated Circuit
IC	Integrated Circuit

CHAPTER I

INTRODUCTION

1.1 Introduction to Vehicle Tracking and Speed Estimation

In recent years, vehicle tracking has been applied in traffic surveillance with the intention of gaining traffic flow information, capturing traffic violations, and classifying vehicles. Vehicle tracking is an undertaking that can open up possibilities for countless other applications. For example, a traffic surveillance camera attached to a traffic light could detect and track vehicles and not be affected by pedestrians. However, we could also configure the image processing system to detect pedestrians and ignore vehicles. It's this versatility that makes image processing a preferred alternative to other detecting methods.

That being said, one aim is to create a system that not only tracks vehicles but subsequently calculates its speed. Velocity has a very simple formula and applying it in image processing should not be that difficult. An object that is tracked will produce a middle point (centroid) which holds the value of the vehicle's midpoint coordinates. Using these coordinate we can determine the displacement, etc. Frame sampling rate, geometric and radiometric resolutions, and distortion amounts of the optical system of the camera affect the precision of the estimated speeds. There are several journals which recall the attempts to develop a speed estimation system. Each was different and unique in its own way. This means that vehicle tracking and speed estimation may have several ways to be determined.

The starting point of many works for traffic surveillance applications is based on the segmentation of the moving object, and for this purpose background subtraction methods are mostly used. For this purpose, each pixel of the successive frame images is subtracted, such that $I(x, y, t) - I(x, y, t + \Delta t)$. The absolute value of this subtraction operation is used. However, since the aim of this project is to solely detect moving vehicles, Optical flow is

used since it specializes in tracking moving objects. In order to eliminate the object shadows, some other operations are often performed on the segmented images. In this paper, we choose to nullify the effects of shadows by running the system when there are no forecast shadows.

1.2 Problem Statement

Traffic surveillance is commonly used to detect and track moving vehicles. However, it does not compute the vehicle's speed which is very important when it comes to road safety. The proposed project, if successful may aid traffic law enforcement to prevent drivers from exceeding the speed limits. Vehicle tracking is very versatile and can be reprogrammed to do additional tasks such as object counting, object distinguishing and more. However, low-tech speed sensors are still being used on highways and traffic lights. The proposed project also uses a more affordable and cheap method compared to high end vehicle tracking systems developed by surveillance companies.

1.3 Aim and Objectives

The development of vehicle tracking and speed estimation for traffic surveillance is the aim of this project. In order to achieve this aim, the objectives have been formulated as follows:

1. To develop a system to detect a moving vehicle.
2. To develop an algorithm that computes vehicle's speed and display it on the output video.

1.4 Scope

1. The system is only tested during the day when there is no whether disruptions such as rain since it may be interpreted as noise by the system.
2. The Matlab/Simulink has been used as a simulation platform to provide easier analysis and alterations to the system.
3. Prerecorded video with 120x160 pixel resolution will be used to minimize the processing power required for analysis.

4. The camera used for the recording of traffic flow is at a fixed height and angle to overcome the depth perception limitation.

CHAPTER II

LITERATURE REVIEW

2.1 Case Study

2.1.1 Tracking an Unknown Moving Target from UAV

Target tracking is one of the most popular vision-based missions a UAV has to solve. Image processing algorithms for such mission can be categorized into two groups. The first group contains ones which operates using given model of the target. It enables direct estimation of relative position and attitude between UAV and target, such as SIFT (Scale-Invariant Feature Transform), SURF (Speeded Up Robust Features) and 3D-shape matching. The other group is composed of ones that run without any modeling such as Camshift, color-based morphology and optical flow. A UAV utilizes optical flow since it enables the detection of objects without prerequisites such as color or shape.

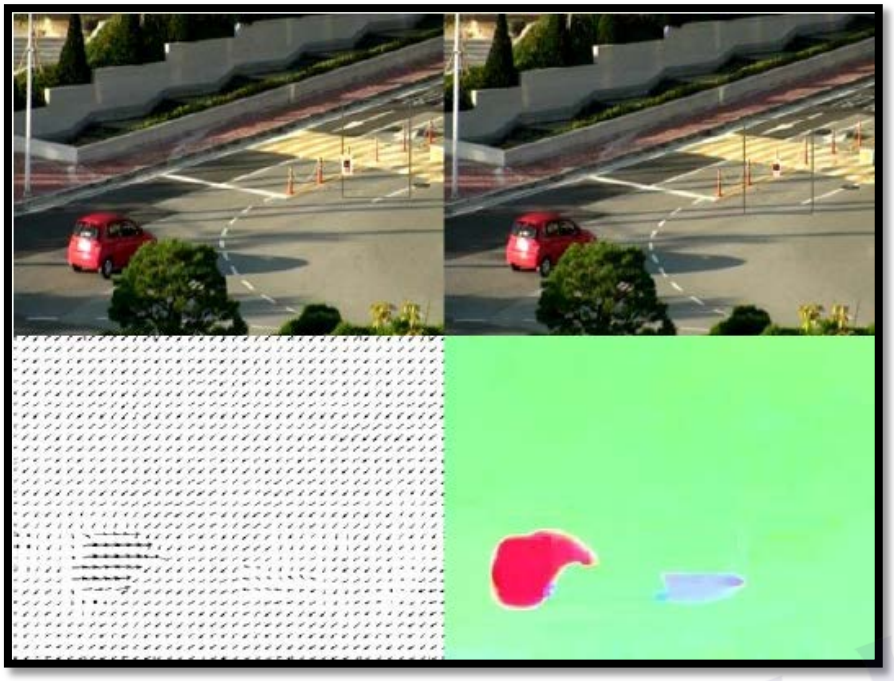

Figure 2.1: For a sequential image from a video of a moving car (top), vector field can be obtained by pyramid Lucas-Kanade optical flow (lower left) and the same information expressed using a color wheel (lower right).

The block diagram in Figure 2.1 shows the UAV utilizing Lucas-Kanada Algorithm to convert the image of the car into a vector field. The generated vector field provides the velocity information of each pixel. This information is displayed in colors by expressing the orientation and magnitude of the vector by varying hue and saturation, respectively.

2.1.2 Real-time Tracking of Round Object

This project uses the autothreshold block to convert an intensity image and convert it into a binary image using Otsu's method. Thresholding is basically a filter that compares each pixel in an image with a value specified by the user. Should the input value be greater than the threshold value, the block outputs a 1; otherwise, it outputs 0. The resulting output would be a black and white binary image, where black are the pixels below the threshold, and white vice

versa. The ball tracking utilizes a Kalman filter which predicts the motion of an object using Bayesian estimation. Figure 2.2 shows the model of real-time ball tracking.

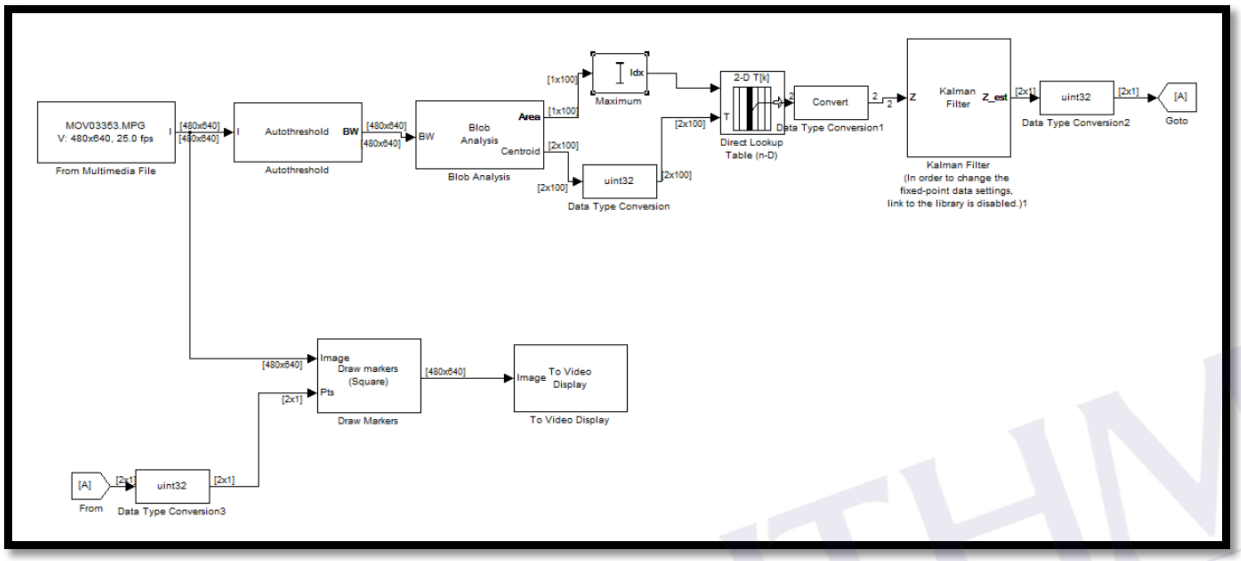

Figure 2.2: Simulink Model of Real-Time Ball Tracking.

2.1.3 Video Stabilization and Motion Detection using the Matlab Video Processing Toolbox

This project implements Matlab Video Processing Toolbox to overcome video turbulence and detect moving objects. The method used is similar to background subtraction, except that instead of a background without any objects, the current frame is compared to the previous frame. Frames which intensity has changed are indicating movement. Figure 2.3 shows the block diagram of the motion detection.

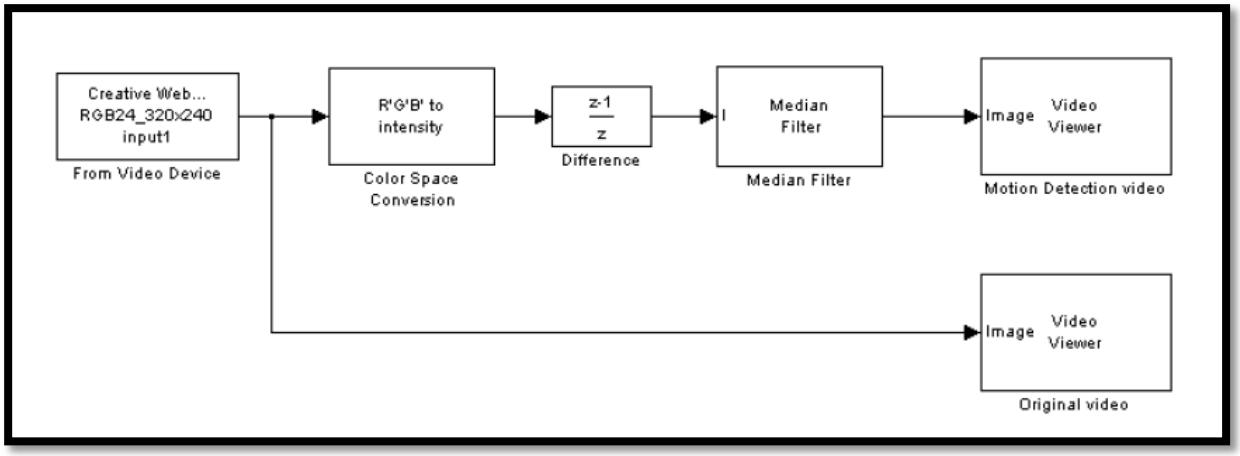

Figure 2.3: Motion Detection using Simulink.

2.1.4 Comparison of Optical Flow Algorithms for Speed Determination of Moving Objects.

This paper compared two optical flow algorithms : Horn-Schunck and Lukas-Kanade. The optical flow is a velocity field in the image which transforms one image into the next image in a sequence.

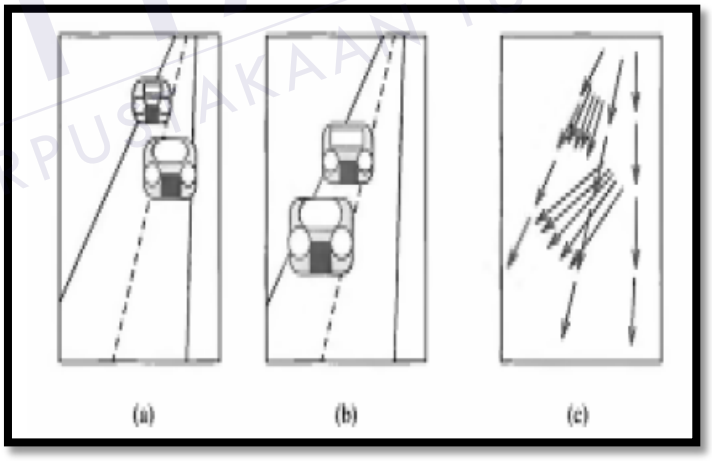

Figure 2.4: Focus of expansion, (a) Time t1 (b) Time t2 (c) Optical Flow.

Horn-Schunck is a global method that estimates everywhere. But Lucas-Kanade algorithm is a local window based method that cannot solve for optical flow everywhere. Lucas-Kanade is more robust to noise compared to Horn-Schunk. The Horn-Schunk method

is also very unstable in case of illumination artefacts in the image sequence. Table 1.1 shows the average angular error for both methods; Table 1.2 shows the PNSR of both algorithms.

Table 1.1: Average angular error.

Angular error Method	Average angular error
Lucas & Kanade	4.3
Horn & Schunck	9.8

Table 1.2: PNSR of various algorithms.

Method	PSNR
Lucas-Kanade algorithm	32.09
Horn-Schunck algorithm	30.71

2.2 Noise Filtering

In image processing, filters are applied to reduce noise and/or to prepare images for further processing such as segmentation. Some noise distributions are very annoying when are involved in intensity changes in video frames. They randomly and sparsely corrupt pixels. Therefore, there is a need for ways to implement smoothing techniques to remove different kinds of noise. Image filtering algorithms have several main objectives such as:

1) Suppression of noise
2) Preservation of edges
3) Removal of impulses

A class of filters that fulfil these requirements is the so called signal filter. Some commonly used signal filters are Gaussian Smoothing, Mean Filter, and Median Filter.

2.3 Image Segmentation

To detect an object, the image is normally segmented into blobs or regions using some common segmentation techniques such as background subtraction mean shift clustering and graph cuts. Segmented regions are then grouped together to represent an object based on some deterministic rules. In tracking algorithm the content of each frame is read and the background is estimated. The unwanted/interested objects are tagged by eliminating the background. Thresholding function is used to convert the grayscale image to binary so that the objects of interest can be highlighted by fixing a threshold limit.

2.4 Closing (Morphology)

Closing is defined simply as a dilation followed by an erosion using the same structuring element for both operations. It is an important operator of the field of mathematical morphology. Like its dual operator opening, it can be derived from the fundamental operations of erosion and dilation. Like those operators it is normally applied to binary images. Closing is similar in some ways to dilation in that it tends to enlarge the boundaries of bright regions in an image, but it is less destructive of the original boundary shape. As with other morphological operators, the exact operation is determined by a structuring element. The effect of the operator is to preserve background regions that have a similar shape to this structuring element, or that can completely contain the structuring element, while eliminating all other regions of background pixels.

CHAPTER III

METHODOLOGY

3.1 Flowchart of Overall Project

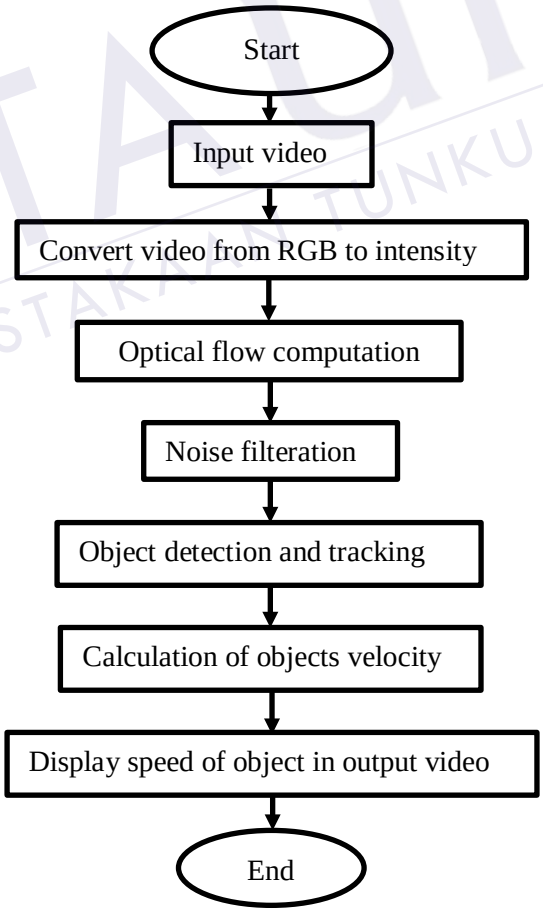

Figure 3.1: Flowchart of the overall project.

The proceeding flowchart shows the chronological order in which this project was completed. Each step is completed by utilizing blocks that are provided in the Image Processing Toolbox.

3.2 Computer Vision System Toolbox Software

Simulink is a graphical extension to MATLAB for modelling and simulation of systems. One of the main advantages of Simulink is the ability to model a nonlinear system, which a transfer function is unable to do. Another advantage of Simulink is the ability to take on initial conditions. When a transfer function is built, the initial conditions are assumed to be zero.

In Simulink, systems are drawn on screen as block diagrams. Many elements of block diagrams are available, such as transfer functions, summing junctions, etc., as well as virtual input and output devices such as function generators and oscilloscopes. Simulink is integrated with MATLAB and data can be easily transferred between the programs. Simulink is supported on Unix, Macintosh, and Windows environments; and is included in the student version of MATLAB for personal computers.

Simulink has a built in Computer Vision System Toolbox that supports a stream processing architecture through blocks. It allows the use of key stream/image processing techniques that are crucial for overcoming noise and differentiating an object from its background. Some of the tools provided in the Computer Vision System Toolbox are:

1) Analysis & Enhancement
2) Conversions
3) Filtering
4) Geometric Transformation
5) Morphological Operations
6) Sinks
7) Sources
8) Statistics
9) Text & Graphics
10) Transforms

3.3 Converting video from RGB to intensity

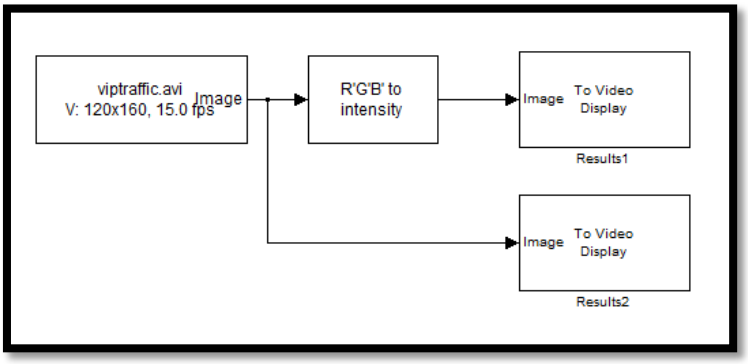

Figure 3.2: Simulink model for RGB to intensity conversion.

The first step for this project is to convert the video we intend to simulate from RGB to intensity. RGB (red, green, blue) are the three colors that can be mixed to become any other colors in the color spectrum. In an RGB video, each pixel is represented by a combination of these colors. Though it provides a more accurate visual representation of the recorded object(s), having to detect 3 colors in every pixel is redundant. Hence, the simplest way is to convert the video from RGB to intensity. What converting the video to intensity does is represent each pixel in the video with a value ranging from 0 to 255. 0 being the color black; 255 being the color white. Any values in-between are shades of gray.

Figure 3.3: The range of intensity values from 0 (black) to 255 (white).

REFERENCES

[1] C. Jay Hyuk, L. Dongjin and B. Hyochoong, "Tracking an Unknown Moving Target from UAV: Extracting and Localizing a Moving Target with Vision Sensor based Optical Flow," Proceedings of the 5th International Coference on Automation, Robotics and Application, Dec 6-8, 2011, Willington, New Zealand.

[2] P. Hidayatul, and H. Konik, "Real-time Tracking of Round Object," 2011 International Conference on Electrical Engineering and Informatics, July 17-19, 2011, Bandung, Indonesia.

[3] B. Wook, L. Tae-Jae, J. Byung-Moon, C. Ji-Ho, C. Jaeil, and C. Dong-II, "Video Stabilization and Motion Detection using Matlab Video Processing Toolbox," 2012 9th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI), November 26-29, 2012, Daejeon, Korea.

[4] Vaibhav Narula, Mukul Sagar and Pranab Joshi, "Real-Time Active Noise Cancellation with Simulink and Data Acquisition Toolbox," 2011 International Conference on Electrical Engineering and Informatics, July 17-19, 2011, Bandung, Indonesia.

[5] A. M. Tekalp, E.F.(1995), "Digital Video Processing. Englewood Cliffs", NJ: Prentice-Hall, 1995.

[6] Different Approaches for Motion Estimation", E.F.(4th-6th June 2009), International Conference on control, automation, communication and energy conservation -2009.

[7] Savan Chhaniyara, Pished Bunnun, Lakmal D. Seneviratne and Kaspar Althoefer, E.F.(MARCH 2008), "Optical Flow Algorithm for Velocity Estimation of Ground Vehicles:

A Feasibility Study", International Journal on smart sensing and intelligent systems, VOL. 1, pp. no. 1.

[8] Lazaros Grammatikopoulos, George Karras, Elli Petsa, E .F.(November, 2005),"Automatic Estimation of Vehicle, Speed from Uncalibrated Video Sequences", International Symposium on Modern Technologies, Education & Professional Practice in Geodesy and related fields, Sofia,03 – 04.

[9] J.L. Barron, D.J. Fleet, and S.S. Beauchemin, E.F.(February 1994), "Performance of Optical Flow Techniques", International Journal of Computer Vision, , vol. 12(1), pp. 43-77.

[10] S. Baker, I. Matthews, E.F.(March 2004), "Lucas-Kanade 20 Years On: A Unifying Framework", IJCV, Vol.56, No. 3, ,pp. 221-255.

[11] Qiaona Pei. "Moving Objects Detection and Tracking Technology based Optical Flow". North China University of Technology, 2009, pp. 11-14.

[12] Lucas B. D., Kanade T. "An Iterative Image Registration Technique with an Application to Stereo Vision," DARPA Image Understanding Workshop, 1981, pp. 121–130.

[13] Silar, Z., and M. Dobrovolny. "Comparison of two optical flow estimation methods using Matlab." Applied Electronics (AE), 2011 International Conference on. IEEE, 2011.

[14] Horn B. K. P., Schunck B. G. "Determining Optical Flow," Artificial Intelligence, vol. 17, 1981, pp. 185–204.

[15] Hageman, L., Young, D. "Applied Iterative Methods," Academic Press, New York, 1981, pp. 18-22.

[16] Eng, How-Lung, and Kai-Kuang Ma. "Noise adaptive soft-switching median filter," IEEE Transactions on Image Processing, vol. 10, no. 2, 2001, pp. 242-251.

[17] Tsekeridou, Sofia, Constantine Kotropoulos, and Ioannis Pitas. "Morphological signal adaptive median filter for still image and image sequence filtering", IEEE International Symposium on Circuits and Systems, vol. 4, 1998, pp. 21-24.

[18] Gonzalez, Rafael C., and E. Richard. Woods, “digital image processing," Prentice Hall Press, ISBN 0-201-18075-8. 2002.