Background Subtraction Using Coplanar Filter and Quadtree Decomposition for Objects Counting

طرح الخلفية باستخدام مصفاة كوبلنر وشجرة التحليل الرباعي لعد الكائنات

Student Name:

Fayez Kamal ALZaghal

Student Number: 401310077

Supervisor:

Dr. Sadeq Alhamouz

A thesis Submitted In Partial Fulfillment of the Requirements of the

The Master Degree in Computer Information Systems

Faculty of Information Technology

Middle East University

October, 2015

بسم الله الرحمن الرحيم

((وَاللهُ أَخْرَجَكُم مِّن بُطُونِ أُمَّهَاتِكُمْ لاَ تَعْلَمُونَ شَيْئًا وَجَعَلَ لَكُمُ الْسَّمْعَ وَالأَبْصَارَ وَالأَفْئِدَةَ لَعَلَّكُمْ تَشْكُرُونَ))

صدق الله العظيم

(سورة النحل/78)

Authorization statement

I Fayez Kamal ALzaghal Authorize the Middle East University
To supply a copy my Thesis to libraries, establishment or individuals.

Signature :

Date:18/2/2016

Middle East University

Examination Committee Decision

This is to certify that the thesis entitled "Background Subtraction Using Coplanar Filter and Quadtree Decomposition for Objects Counting" was successfully defended and approved in 2016.

Examination Committee Members **Signature**

(Supervisor)

Dr. Sadeq AlHamouz
Head of Computer Information System
Middle East University

(Chairman of Examination Committee)

Professor Ahmad Kayed
Dean Faculty of IT
Middle East University

(External Committee Members)

Dr. Suhail Owais
Head of Computer Science
Applied Science University

Acknowledgment

In the first, my thanks are hereby extended to my God, then to **Dr. Sadeq Alhamouz** for his supportive and helpful supervision, as well as for assisting a student in every step of the project, and for providing important information and basics, which was very important for the successful implementation of the project. Further thanks are extended to everyone especially **Dr. Hebah Nasereddin** who helped me develop my understanding of the various nuances of the project and for everyone who believes that the knowledge is right for everyone.

Dedication

To my father who taught me how to find my way through man's most challenging hardships, who support me to the end, and he is my idol.

To my mother who dedicated her life to watch over me, Who sacrificed for us, mom you're the best .

To my brothers and sisters with whom I shared with the joy of life.

To my friends whom always with me, support me to the end.

To every one who helped me morally .

To my friend, my lifetime companion; Imad Khuffash.

To the one who guided me toward success in my academic advancement Dr. Sadeq Alhamouz

List of Contents

List of Figures

List of Tables

Background Subtraction Using Coplanar Filter and Quadtree Decomposition for Objects Counting

Student Name

Fayez Kamal Alzaghal

Superviser

Dr. Sadeq Alhamouz

Abstract

Traditional background subtraction algorithms are used mainly to discover objects in images by subtracting them from known background images for same scenes excluding these objects. However, these traditional algorithms fail in detecting all edge pixels, which is in turn influences the accuracy of resulted detected objects. Therefore, this thesis introduces an enhancement for the traditional background subtraction algorithms, considering applying two techniques within segmentation tool; coplanar filter; to improve the detection of all edge pixels, and Quadtree Decomposition; to divide images into homogenous blocks. Both algorithms; the enhanced, and traditional are then applied to design a car tracking system using MATLAB to detect and count the number of cars in a specific street. The number of detected cars resulted using each algorithm is compared later with the actual number of cars in that street for performance evaluation purposes.

The evaluation is conducted based on detecting and counting the number of cars both in; video frames within the online stage or uploaded images from a dataset within the offline stage. After that, comparing these frames (images) with a background image for a street, which is devoided of cars. The threshold of the proposed system will be adaptive over each segment of the image, normally threshold for traditional background subtraction is 0.5 where any pixel value greater than 0.5 assumed to be white while lower than 0.5 is assumed to be black. This research results illustrate that the enhanced background subtraction algorithm outperforms the traditional algorithm in counting the number of cars in all frames. This system achieved 47.01% average accuracy rate for the traditional background subtraction algorithm and 81.19% average accuracy rate for the enhanced algorithm.

Keywords: Background subtraction; coplanar filter; Quadtree Decomposition; car tracking system; video frames.

لعد الكائنات طرح الخلفية باستخدام مصفاة كوبلنر وشجرة التحليل الرباعي

اسم الطالب

فايز كمال الزغل

المشرف

د.صادق الحموز

ملخص الرسالة

يتم استخدام خوارزميات طرح الخلفية التقليدية لايجاد الأجسام في الصور بالاعتماد على طرحها من صور خلفية لنفس المشاهد دون هذه الاجسام. ومع ذلك, فشلت هذه الخوارزميات التقليدية في تحديد جميع حواف البكسل للصور، حيث إنّ هذا بدوره يؤثر على دقة النتائج. وبالتالي، فإن هذه الرسالة تقدم تعزيزا للخوارزميات التقليدية بالاعتماد على تطبيق تقنيتين : فلتر متحد المستوى (coplanar) لتحسين إمكانية الكشف عن كافة حواف البكسل ، وطريقة Quadtree لتقسيم الصور لكتل متجانسة. كل من الخوارزمية المحسنه والخوارزمية التقليدية ثم تطبيقهم لتصميم نظام تتبع وعَد السيارات في شارع معين باستخدام MATLAB. ومن ثم مقارنة عَدد السيارات الذي تم حسابها بكل خوارزمية مع العدد الفعلي للسيارات في هذا الشارع لتقييم أدائهم.

تم تنفيذ التقييم من خلال ايجاد وإحصاء عدد السيارات في لقطات الفيديو في مرحلة ال (online) او في صور تم تحميلها من قاعدة بيانات في مرحلة ال (offline)، بالاعتماد على مقارنة هذه اللقطات او الصور مع الصورة الخلفية للشارع بدون سيارات.وتم استخدام ال(threshold) لتحديد الفرق بين الصور,ويتم عادة استخدام ال(threshold) في الخوارزمية التقليدية بقيمة 0.5 ,بحيث اذا كانت نتيجة طرح البكسل اكبر من (threshold) فان البكسل يكون باللون الابيض,اما اذا كانت اقل من ال(threshold) فان قيمة البكسل تكون باللون الاسود, توضح النتائج أن الخوارزمية المحسنة تفوَقت على الخوارزمية التقليدية في احصاء عدد من السيارات في جميع اللقطات. حقق هذا النظام 47.01٪ متوسط معدل دقة للخوارزمية التقليدية و81.19٪ متوسط معدل دقة للخوارزمية المحسنة.

الكلمات المفتاحية :طرح الخلفية ; مصفاة كوبلنر ; شجرة التحليل الرباعي ; نظام تتبع السيارات ; اطار الفيديو.

LIST OF ABBRIVATIONS

FCH	Fuzzy Colour Histogram
FD	Framework Difference
RABU	Remembrance Adaptive Background Update
RGB	Red Green Blue

Chapter One

Introduction

Chapter One: Introduction

1.1 Introduction

Using computers for image processing purposes provides more powerful results as well as more implementation flexibility to those who concerns in image processing applications rather than using electrical resources. This can be explained by the simplicity and the easiness of applying modifications on its operations when it is necessary to the computer programming software used for image processing purposes in quite faster manner.

Generally, real time division and tracking of moving objects in video frames is a basic and critical issue in various computer vision systems, such as; human machine interfaces and visual surveillance. The most common used technique to recognize and detect moving objects is the background subtraction algorithm which based on comparing the current frame with a background frame, the current frame pixels-which diverge considerably from the background-are considered as the moving objects, those detected pixels are then processed to track and localize objects. Due to the importance of background subtraction algorithm in tracking moving objects, it must be ensured that all pixels related to the moving objects are correctly and precisely detected and extracted without losing any pixel. Practically, maintaining non-loosing of pixels during this algorithm evaluation is quite obvious challenging. (Vibha. et al, 2008; Cheung and Kamath, 2015)

In practice, background subtraction algorithms have a similar flow diagram with four main stages; preprocessing, background modeling, foreground detecting and data validation. The first stage includes various image-processing tasks to convert the input video into a certain format; however, this can be processed via consequent stages. The second stage depends on using the preprocessed video frames to compute and revise a certain background module which offers a numerical explanation of the whole background frame. The third stage signifies the video frame pixels which cannot be sufficiently expressed using the background model. Now, the resulted pixels are considered as a binary applicant foreground mask. The final stage evaluates this resulting mask, removes pixels which are not related to any real moving object and shows the final required foreground mask. After that this mask is used to compare it with real moving objects to make the right decision later. (Cheung and Kamath, 2015).

Various types of filters and models have been proposed and used to enhance and ensure the detection of all foreground pixels, such as Weiner filter, median filter, Kalman filter, histograms and Gaussian mixture (Cucchiara. et al, 2003). However, those filters and models have fixed implementation parameters where this is not adequate for scenes, as traffic intersections this include moving objects at various speeds (Cheung and Kamath, 2015).

1.2 Problem statement

Practically, object extraction algorithms are utilized to suppress the background of a certain video scene in order to discover the frame objects. The aforementioned traditional algorithms subtracting and comparing the existing frames from the background frames where the remaining pixels are treated as foreground. However, with the use of those algorithms, the majority of edge pixels cannot be detected, where this in turn effects on the accuracy of the results and the effectiveness of those algorithms. Therefore, the motivation of this research work is to adopt an enhancement for these traditional background subtraction algorithms as an effective contribution to overcome the aforementioned limitations, this adoption is expected to detect the whole edge pixels which in turn results in increasing both the detection accuracy of counting the cars number and the efficiency of using such algorithms for that purposes especially in hot areas like hospitals, airports. This enhancement over the traditional algorithms will be done by applying some modifications using segmentation tool within both methods; quad tree decomposition and coplanar filter which will be briefly explained in the methodology chapter later. Since background subtraction is a critical technique used in image processing field due to its huge role in the detection of moving objects' applications it gained its importance, and using these methods of quad tree decomposition and coplanar filtering in this research work will even gets this algorithm additional feature of providing an accurate results after achieving the aim of this research in enhancing its resulted accuracy throughout the use of extracting better results of the enhanced algorithm that applies these techniques together.

1.3 Objectives

The main purpose of this research is to enhance the traditional background subtraction algorithms, then apply the enhancement algorithm for the purpose of designing a car tracking system using MATLAB software. This system however will be designed as a practical proof for the capability of the enhanced adoptive algorithm in tracking, and counting the number of cars that enter the gate of a certain place. Moreover, proposing such application system will show the importance of this adoptive enhancement algorithm in building such systems that benefits everyday use in life which Strengthens our research objectives as it is useful and applicable enough to be achieved and discussed. As earlier stated the background subtraction algorithm will be enhanced based on applying segmentation process within the use of quad tree decomposition, and coplanar filtering to improve maximizing the detection of pixels' edges to include all pixels edges unlike the traditional algorithm. Listed below are the research objectives to achieve the main aim of this research:

- Explore the main issues, and concepts of image processing techniques.
- Explore the main concepts, benefits, and limitations of the traditional background subtraction algorithm.
- Explore the use, and application of segmentation, using quad tree decomposition.
- Review some of the recently published works concerning the use of the background subtraction algorithm in car tracking systems, with discussing their methodologies, outcomes, and limitations.
- Discuss, and evaluate the implemented system, based on applying it on various images.

1.4 Significant of the thesis

This research significance comes from the proposed algorithm significance; as discussed later in sections (1.2), and (1.3), the background subtraction algorithms-the traditional algorithms- are significant in detecting the moving objects, but for improving purposes the researcher decided to adopt an enhancement for those algorithms in addition to propose an applicable system using the enhancement algorithm to profit from the efficient use of this algorithm after obtaining an accurate results due to the use of both quad tree decomposition and coplanar filtering techniques in segmentation process to detect all pixels' edges which increases the accuracy of detecting the moving objects. For that reason the researcher tended to use this enhancement algorithm in implementing a tracking car system to produce an applied proof of the importance of using such algorithms in the world of image processing and computer vision. Since this research explores the implementation of an advanced background subtraction algorithm based on digital image processing system it uses MATLAB to achieve that goal. This implemented system is important for daily life use because of its accurate results in counting the number of cars that enter a specific area (university gate, intersections, street etc….), and providing video observation, analyzing traffic, navigating the vehicle and even much more which can be important also for future works and future systems and applications based on its implementation idea. What distinguishes this system which uses this research proposed algorithm is its ability in providing a qualified, inexpensive, robust, and reliable system for monitoring traffics, to count, detect, and track cars.

1.5 Research Methodology

In this research, an advanced background subtraction algorithm based on digital image processing system using MATLAB software will be implemented, tested, and estimated to count the number of cars. The proposed algorithm will initially consist of two components; camera, and MATLAB software. Furthermore; It will be divided into two stages: online, and offline. In the online stage; the camera that will be located in a specific point, will records a video for the traffic, while this camera will be connect with a pre-written codes in MATLAB which will be installed on a personal computer. In the offline stage, images will uploaded from a database instead of using the camera, and then the algorithm will take place to detect the moving objects in them. After that; for each stage the evaluation part will take place for some metrics to estimate the efficiency of both the traditional and the improved algorithm in detecting the moving objects. For evaluation measures, the researcher selected the accuracy and efficiency of detecting all pixel's edges the as result to those previously mentioned the moving objects will be detected more precisely. All the research work's results will be illustrated, demonstrated, and compared for both algorithms till the objectives of this research achieved effectively.

1.6 General Outline

This research is consisting of the following chapters:

* **Chapter One: Introduction**.

The first chapter includes the following; the main aims; problem statement; the importance of the research; research methodology, detailed information about the traditional background subtraction algorithm.

* **Chapter Two: Literature Review.**

This chapter reviews the works of a number of researchers, who have studied the traditional background subtraction algorithm, including their results, and a brief description of their methodologies.

* **Chapter Three: Methodology**

This chapter explores the description, and implementation of a new mechanism to improve the traditional background subtraction algorithm, and then apply it to design a car tracking system with the use of MATLAB program**.**

* **Chapter Four: Testing and Discussion.**

This chapter introduces a detailed discussion of the study testing aided with the required diagrams and graphs.

* **Chapter Five: Conclusion and future works.**

This chapter concludes the whole work that has been introduced and achieved during the research and the possible future works for the project.

Chapter Two

Literature Review

Chapter Two: Literature Review

Raghtate, &Tiwari (2014) introduced an algorithm to count the number of cars on the street using background technique; the output of this process is a binary and a clearer image of the object. To get a clearer image of the foreground object, a morphological closing operation is required to be done on the binary image, by computing the difference between the basic background and foreground picture, after the processing, reduce noise and shadows, then use morphological gradient operation, that uses median filter, without disturbing the object shape, they used MATLAB tools to subtract the picture, for example; Deblurring Images using a Wiener Filter.

Sajjad (2010) stated that preprocessing stage is a set of processes that can be applied on images in order to correct, standardize and enhance them, as well as improving their quality. It is considered as an important stage in all the computer vision systems. The main pre-processes that can be applied on images are: resizing, color space converting, and filtering. In the resizing process, the image size is adjusted since images that have large sizes may slow down the system. In the process of converting the color space, colored images that are in the Red, Green and Blue (RGB) color space are converted into grayscale images. In the filtering process, various types of filters are applied on images in order to remove noises, remove the blurring effects, sharp images and enhance the image edges. The main types of filters that can be applied are the high pass and low pass filters that remove the unwanted low frequency and high frequency components, respectively.

Samuel and Liu (2010) argued that the background subtraction algorithm is a widely used technique in the fields of computer vision and processing of images. It is used to separate the required foreground objects from backgrounds in both images and videos. It depends on comparing an observed image with an estimated one using a specific threshold in order to specify the locations of the desired objects.

Hwang et al (2009) represented methods that can adopt the background model to various situations, so the system can detect the moving and stopped objects, but the most widely used before Mixture of Gaussian to detect the moving object, by using background subtraction algorithm, to allow the system to detect the moving objects they adopted Mixture of Gaussian

Model in urban transit. The parameter should be adapted in various situations. They train the model and get the adaptive parameter by using the time gap between moving and stopped objects.

Three main steps in background subtracting are existing; the object detection, objects tracking, and post processing. Though, the background subtraction method is simple to detect moving objects, several difficulties arise, these difficulties are; noises, fluttering objects, dust particles, and illumination changes.

Sigari (2008) mentioned some weaknesses of the running average method and standard background subtraction. Then, a fuzzy approach for background modeling and background subtraction is proposed. For fuzzy background modeling, fuzzy running average is suggested.
A compare between running average method and background subtraction with their fuzzy approaches in the real world are integrated, so that both the classic and fuzzy algorithms were used in vehicle detection application. Experiments have been done in the evening, because of extensive illumination changes and high vehicle traffic density. Experimental results show that fuzzy approach is 6% more accurate than classic approach. However, fuzzy vehicle detection is 12% slower than classic vehicle detection.

Vibha et al (2008) dealt with presenting a background registration technique to detect moving objects. To count the number of dynamic objects proficiently on using combines simple domain knowledge about object classes with time domain statistical measures to identify target objects in the presence of partial occlusions and ambiguous poses, and the background clutter is effectively rejected.

Rostamianfar et al (2006) introduced a system for detecting and tracking pedestrian as well as vehicles in traffic intersection by using background subtraction method, then update and use it in real time situation, the system was tested in different and difficult condition. The system efficiently detects and identified vehicle tracking. The job done by blob analysis technique for extracting binary image facilitates pedestrian and car detection. Processing blob's information of relative size and location leads to distinguishing between pedestrian and car. Applying temporal analysis techniques and moving object detection methods improves system versatility to detect and recognize waiting and moving pedestrian and car. To enhance system robustness to scene changes and reduce the error rate an innovative method of Remembrance Adaptive Background Update (RABU) is presented, the proposed method scans the scene changes and includes relatively unchanged field to background update.The system is able to report events, and this can be extended to control the traffic signals in intersection as a replacement for existing loop detectors and push buttons. The system is fully automated and self-triggered for background update, and this makes it smart and adaptive to scene changes.

In this work, (Tamersoy, 2009) each input image is compared with an image that represents the streets without cars. In other words, all the test images are subtracted from that image to count the number of cars that entered the street. Supposing that the input image is denoted by I(x,y,t), while the testing image (image of streets without cars) is denoted by B(x,y,t) where x and y represent the coordinated of pixels and t represents the estimated time, then the background subtraction formula that can be used to compare the pixels of both images and determine if they are a foreground or background can be represented as follows using a specific **threshold** value (Th).

$$If\ |I(x,y,t) - B(x,y,t)| > Th, then\ the\ pixel\ is\ \text{foreground}$$

$$Else \quad the\ pixel\ is\ background \quad (\text{ TAMERSOY}, 2009)$$

The following figure 1 shows an example of the proposed algorithm where each pixel in the input image is subtracted from its corresponding pixel in the training image, after that, the absolute value of the result is compared with a threshold. If the subtraction result is bigger than the threshold, that pixel is considered as a foreground (object), else, it is considered as a background (Tamersoy, 2009).

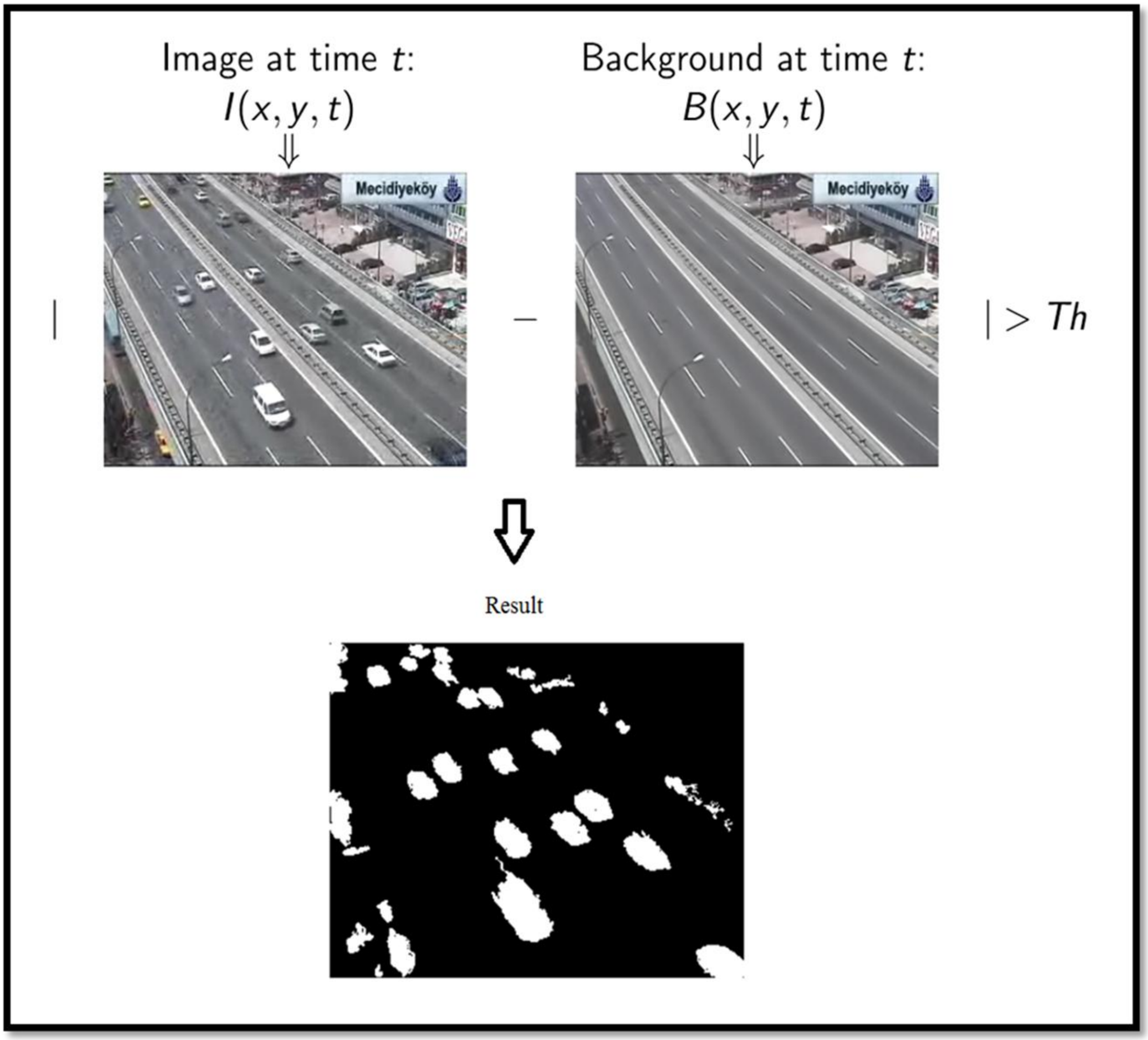


Figure 1: background subtraction algorithm (Tamersoy, 2009).

Nagendran et al (2014) proposed a method for efficiently tracking the moving objects in the captured video that taken via locomotive camera in complex views. The sequence of the video may include highly dynamic illumination modification and backgrounds. The suggested method contains four steps. The first step is stabilizing the video via affine transformation, the second step is selecting frames in an intelligent way to extract just those frames, which have a large change in the frame content. Through this step the computational and complexity time reduced. The third step is tracking the moving object via “Gaussian mixture” model and “Kalman filter”. The fourth step is recognizing the moving objects via performing Bag of features.

Anitha. et al (2013) proposed an illegal immigrants, conflicts and monitoring military are areas depends presently on manpower and technology, but the “automatic monitoring” has been improved to eschew the errors of the potential human, which can be caused via various reasons. This conducts an “automatic recognition” of the object, which uses “a neural network” in the process of recognizing the extracted object, and image processing in the process of detecting and extracting the moving object among a limited area. The suggested system still receives the captured images every two seconds via the monitoring camera, which observes a restricted zone, like international limit crossings, buffer area, or any observed area where the motion in that region is to be discovered. The implementation consists of three phases, the first phase is the detection process for the moving object which is done via the variance of the value of image pixel, a peremptory rule that used in determining the “moving object”, and the background reparations. The second phase is the process of extraction the detected object, which is done via a group of peremptory rules to find the pixel differences among the detected object(s) image(s) and removing phantom objects, which probably have been gained in the first stage. The second stage has more of the processing for the “extracted object image”, like framing, scaling, and squaring the image to previously defined size in the process of preparing the next stage. The final stage is the process of recognizing the extracted object via a “supervised neural network” depends on simple algorithm, but it is an effective algorithm for the “back propagation learning”. The suggested system gives solutions for the monitoring secured regions problem, the movement across the region detection, moving object’ extraction, and the object’ recognition. The Figure (2) explores the phase of object recognition.

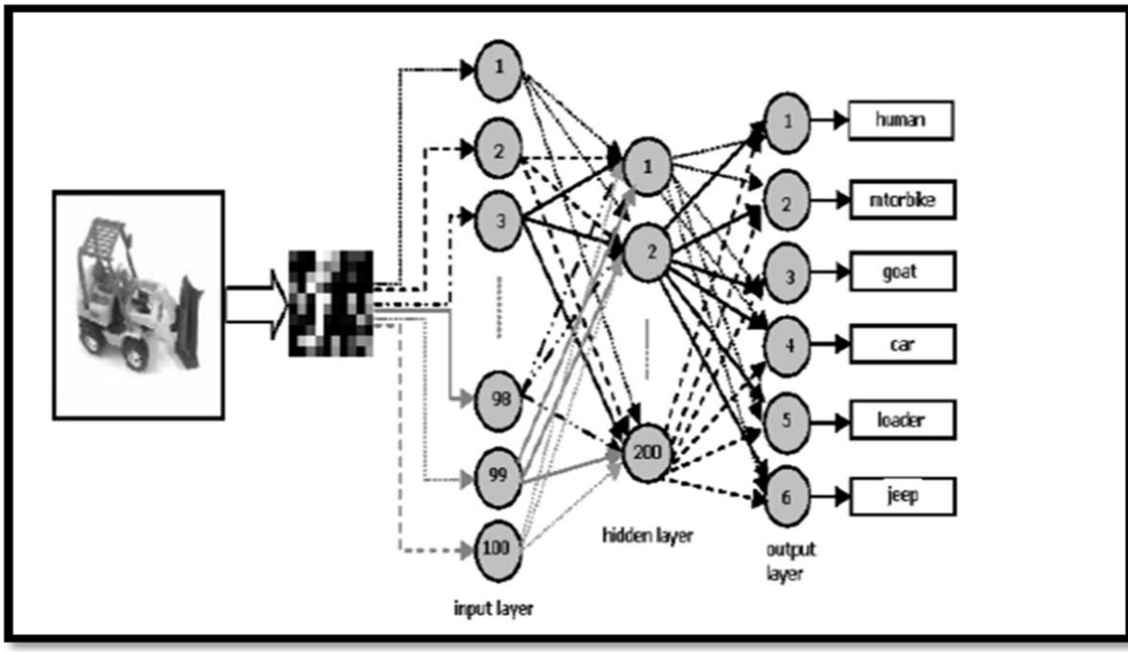


Figure 2: the phase of object recognition, (Anitha. et al, 2013).

Mahamuni et al (2014) said in their paper that the methods of the background subtraction are vastly exploited in the detection process for the moving object in videos in abundant applications, like video surveillance, traffic monitoring, and human movement capture. Some of the most challenging and special sides of such ways. The authors suggest a general aim method, which includes statistical assumptions besides the knowledge level object of animated objects, in the processing stage for the former frames, the shadows and the visible objects (ghosts) are obtained. The moving objects' pixels, shadows and ghosts are processed in a different way in order to provide a selective update based on object. The suggested approach takes advantage of information of gray color for "both background subtraction" and used it in the process of improving the segmentation of the object. The approach evidences flexible, accurate in terms of the accuracy of the pixels, and fast. The background subtraction implementation is achieved in two ranges code, which written via MATLAB, after that, sets of Simulink blocks have been used. The next figures show the block diagram of the system, and the model of the "Simulink Software", which used for the motion detection respectively ,the Figure 4 show the block diagram of the system ,and the Figure 5 Simulink software model.

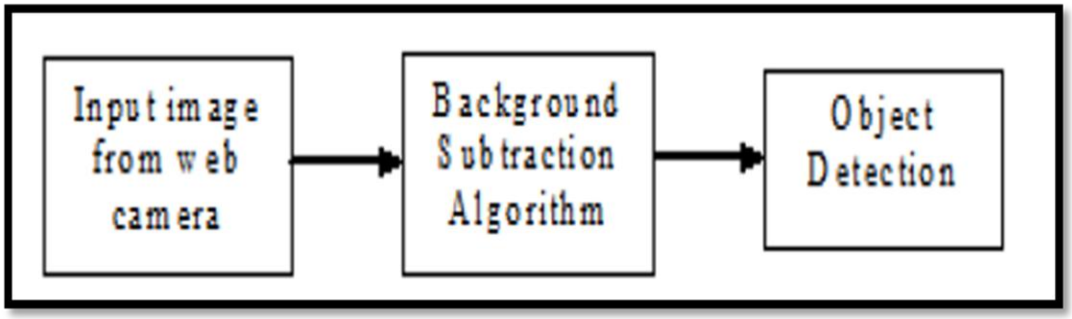


Figure 3: the block diagram of the system, (Mahamuni. et al, 2014).

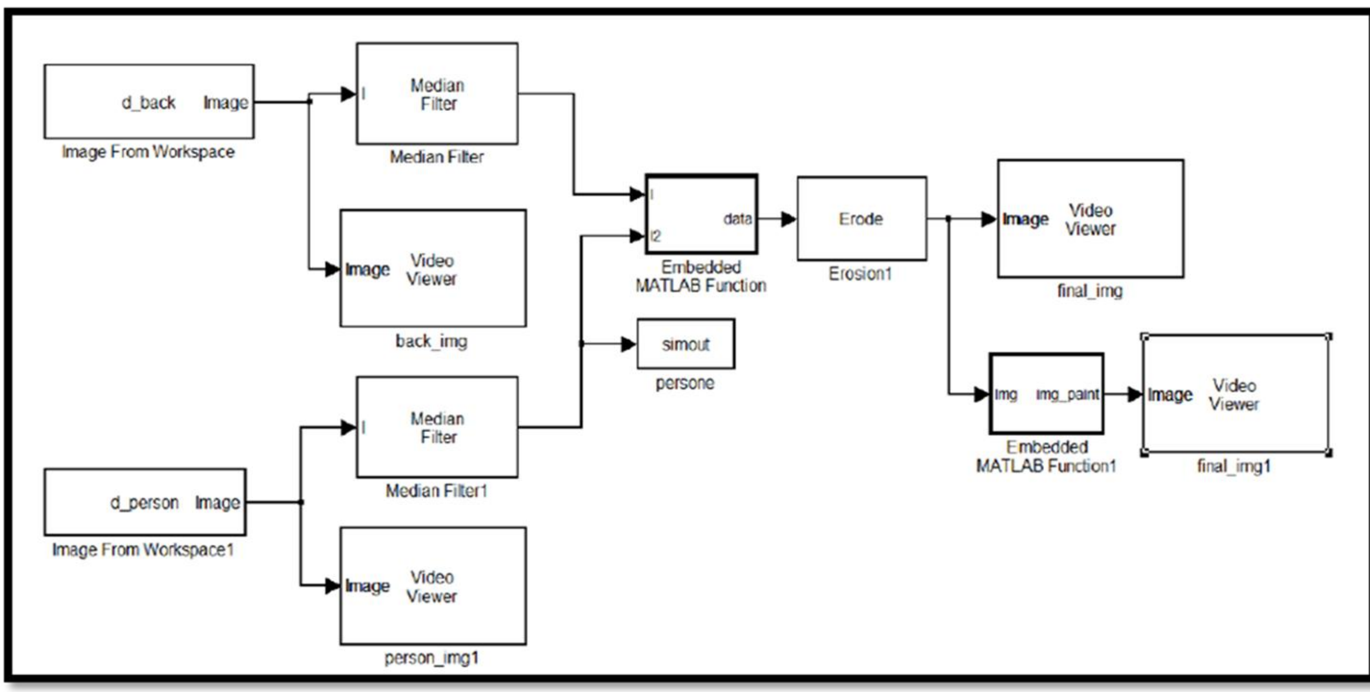


Figure 4: Simulink software model,(Mahamuni. et al, 2014).

Anchez-Ferreira et al (2012) defined the "Temporal differencing method" as a method that use two or three close frame depending on the image of the time series in subtracting and obtain difference images. The work of this method is very comparable to the method of "background subtraction", and after the process of subtraction for the image, it provides information about the moving target via the "threshold value". It is concerns as easy and simple method from the implementation side. But it has a/ high ability to adapt to dynamic view modifications. However, it mostly fails in the process of detecting whole pixels of some kinds of the moving objects.

The authors Augustin. et al (2011) explored that “Background Subtraction method” is commonly used the process of segmentation in the constant images. It detects moving areas via subtracting the present image from a “reference background image”, which is created in the initialization period via “averaging images over time”. The main idea of the method of the “background subtraction” is firstly initialize a background, and after that detecting the moving object via subtracting the present frame of the moving object from the background frame to detect the object. Although, this technique is easy and simple to recognize, and carefully extracts the properties of the target data, it is critical to the modification of the external environment, so that it is viable to the background condition.

The authors Moeslund and Hiton (2006) discussed the Optical flow method and said that is a method uses in the process of detecting the motion region in the image sequences via using the moving target of the features, which modified with the time. It provides better execution below the moving camera, but this method is very complicated and complex computation, and it requires particular hardware support too, so that it is hard to satisfy the “real time video processing” requirements.

The authors Tumul and Nagalaxmi (2014) illustrated in their paper that the “background subtraction” is a significant and challenging task in the dynamic views. They suggest in their paper an effective system for the motion detection based on the method of “background subtraction” via morphological processing and “fuzzy color histogram”. There are two methods have been used in an effective way in the detection of the object followed by persons counting, and then compare these executions based on precise estimation. Filtering and morphological processes have been used in an effective way in the dynamic texture views for undesirable pixel elimination in the background. They out a background subtraction method for provisionally dynamic texture views via “clustering-based feature” called by “Fuzzy Color Histogram” (FCH), which has a capability of luxuriously attenuating color differences, which generated via background movements while still shed light on the detection of the moving object. The experimental results explain that the suggested method is efficient for the process of motion detection in the system based on the "background subtraction" method via "Fuzzy Color Histogram" and the morphological processing, as compared to various competitive procedures. The Figure 5 shows the model of object detection.

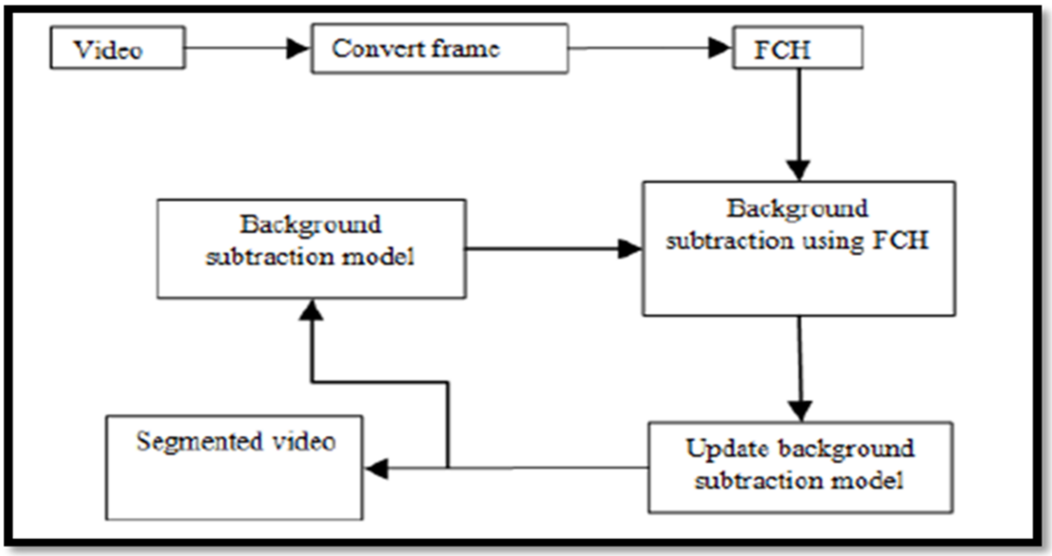


Figure 5: Model Of Object Detection, (Tumul and Nagalaxmi,2014).

Chapter Three

Methodology

Chapter Three: Methodology

3.1 Introduction:

Explaining for the method structure is applied to conduct the outcomes and performance results within procedures based on logic as well as a well-structured flow of methods and approaches. Similar to whichever research procedures that is developed for some purpose, this work confirms the standards of the research methodology to process and estimate the phases of the research which starts from forming the problem of the research till outcomes evaluating and testing.

In this chapter, some important sections are conducted such as flowchart of the research, the research design, methods of the research, approaches and used algorithm until the summary of the chapter concluded. However, the Research methodology was designed and analyzed based on logical and academicals methods to expand the research work and results properly.

3.2 Design Approach

In this research, an advanced background subtraction algorithm based on digital image processing system using MATLAB is implemented, tested and evaluated to collect the number of cars. This system is to monitor the traffics, and to count, detect and track cars. Figure (6) shows the algorithm flow chart for the background elimination, and Figure (7) shows the hole system which will be explained in page (22).

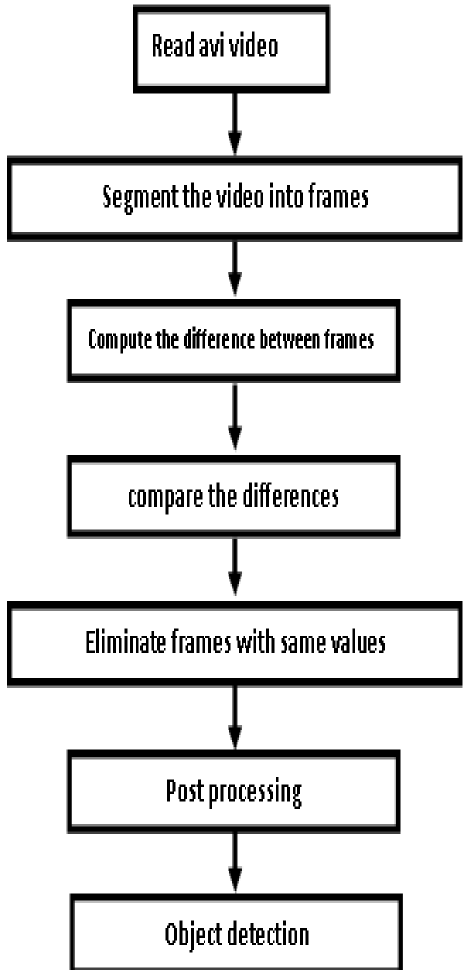


Figure 6: Diagram for the background elimination.

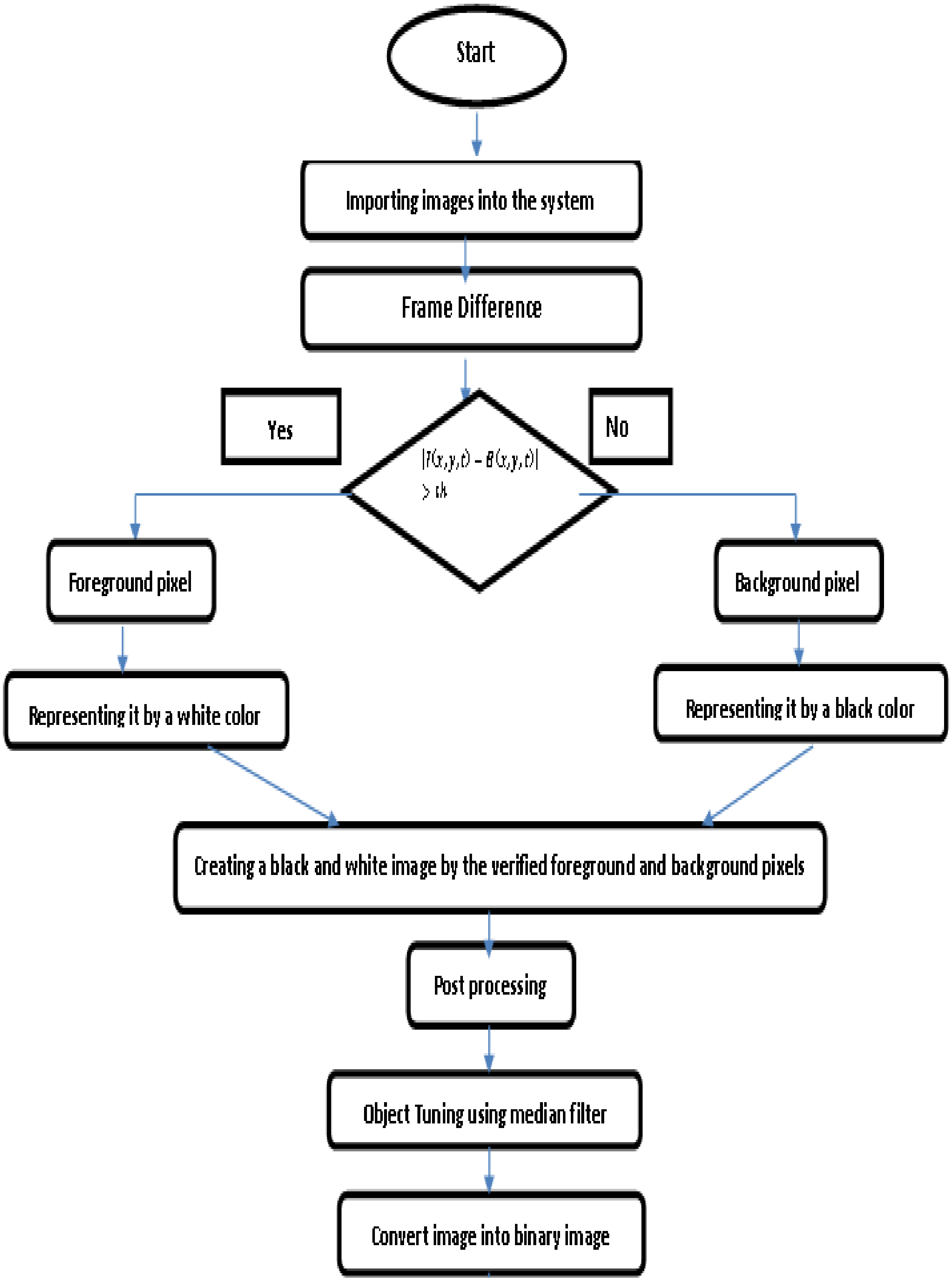


Figure 7:system design flow chart.

.

3.3 Methodology:

Determining moving objects in a video chaining is a decisive and essential task in many “computer-view applications”. An effective algorithm for finding out a moving object utilizing background removal technique is proposed in this research. While the algorithm used in this research is the background subtraction algorithm, where the researcher adopted enhancing this algorithm by using segmentation to the image frames, then applying both the quad tree decomposition and the coplanar filter to get higher accuracy rates for detecting the moving objects. To do so, two essential phases are integrated, the first one is the pre-processing phase, where the morphological gradual operations with median filter are applied to eliminate the shadow and noise areas, which are exist in the “moving object”. the second is post processing phase. In the post processing phase, Quadtree Decomposition and Coplanar Filtering was applied. In the following sections these two methods will be explained briefly.

3.3.1 Work principle:

This research principle is based on the methods used in this research to obtain the desired results. However, In the proposed research system constant camera is used to record video for a certain street. Then, the video is segmented into a number of frames, after that these frames are entered in this system to process them by the background subtraction. The resulted pixels from subtraction is then compared with respect to a given threshold according to the condition says; if the result obtained from the comparison process is greater than zero, then it is represented with white color and given 1 number, otherwise, if it is less, then it is represented in black color and given zero number. This results in obtaining two images; one with white color, while the other with black color. The median filter will be processing next, while both quadtree decomposition and coplanar filter will be executed right after. The obtained result after all the aforementioned processes is the detection of object (car) and then, counting the objects (cars) will be performed.

3.3.2 Background Subtraction:

The background subtraction is performed on an image sequence taken with a static camera to detect changes as object background subtraction compares image with background model. It always assumes that no object appears in pictures once building the background model) (Shivaji, Satish, & Hridyanath, 2014). This technique is utilized in many fields, for example, in the monitoring system to segment a moving object just in an effective way.

The steps that must be followed to implement the background subtraction algorithm are:

- Learning background step: this step gets ten background frameworks via camera and compute the variance (σ) and mean (μ) using the next equations mentioned in (KaewTraKulPong and Bowden ,2004), where x_i is background number.

- $\mu = \frac{1}{n} \sum_{i=1}^{n} x_i$ ……………………… ……………………….. (1)
- $\sigma = \sqrt{\left(\frac{1}{n} \sum_{i=1}^{n} (x_i - \mu)^2\right)}$ …………………………………….. . (2)

- The supposition is that the background value was normal distribution.

3.3.3 Algorithm

The research algorithm tends to directly convert the "RGB" images to binary image, then it uses the subtraction operator which takes two images as input to produce the third image, the third image however is produced after subtracting the second image pixel from the first image pixel by use of the equation number (3), this equation is representing the difference between two images.

diff(x,y)=i2 (x,y)-i3(x,y) ………………………………………………………………. (3)

First of all, the image converted into binary image with the use of threshold value, then the resulted image is processed by morphological operation, Thresholding the difference between two consecutive input frames is the basic concept of change detection based segmentation. It works as follows:

Input: All previous frames are stored in a memory buffer and the current frame in video is (Fi)

1. Take ith frame (Fi) as input image.
2. Take $(i\text{-}3)^{th}$ frame (F_i-3) from the image buffer.
3. Perform Frame Differencing Operation on the Ith and $(i\text{-}3)^{th}$ frame. The resultant image generated is represented as: $\mathbf{diff_i = F_{i-3} - F_i}$ **.**

The image buffer is generally a temporary buffer used to store some of previous frames for future use.

The technique is to remove the limitation to detect slow moving object, which makes it speed independent of moving object and more reliable. After using the frame differencing operation the binary threshold operation is performed to convert the different image into a binary image with some threshold value and thus the moving object is identified with some irrelevant non-moving pixels due to flickering of camera. The binary image (F_{bin}), in which the pixel corresponding to moving object is set to 1 while the rest is treated as background which sets to 0.

This Threshold technique work as, a brightness Threshold (Th) is chosen with the diff (x,y) to which threshold is to be applied:

if diff(x,y) >= Th then

$F_{bin}(x,y) = 1$ //for object

else

$F_{bin}(x,y) = 0$ //for background

This assumes that the interested parts are only light objects with a dark background. But for dark object which having light background we use:

if diff(x,y) <= Th then

$F_{bin} = 1$ //for object

else

F_{bin} = 0 //for background

The threshold is not fixed it can vary according to our perception. The use of Threshold (Th) is just to separate the objects' pixels from the background (Shivaji, Satish, & Hridyanath, 2014).

3.3.4 Modeling and Architecture:

In different real-time applications where the used camera is constant such as; human machine interfaces and visual surveillance. Some techniques utilize "global motion estimation" and comparison to make up the background change because of the motion of the camera. The presumption in the current algorithm is; the background is fixed for the clips of the video. The steps of the algorithm to remove the background are as next:

1. Read the "video' clip" and transform it to frameworks.
2. Compute the difference between frameworks (Fi and $Fi+k$).
3. Comparison between these differences.
4. In the difference framework, remove pixels, which are having the same values.
5. Execute the stage of post processing on the obtained image in the previous point.
6. Detect object.

The following are the modeling and architecture phases where in each subsection a brief abstract are integrated to explain the detailed steps occurring in each phase till the desired results obtained. However, these phases are;

3.3.4.1 Framework difference:

Framework differences are calculated by computing the difference within frameworks order, but if the clips of the video have "slow-moving objects", then it will leads to computational complexity. Consequently, the research work considers the dissimilarity among the frameworks like standard periods taking it for granted an amount equal to some integer k. The conducted framework difference (FD) can be defined with (n/k), within the existence of n frameworks. One important piece of information is that the FD follows Gaussian Distribution, in which a function presented as P(FD) can be defined with probability density function of framework difference. However, the variable μ which included in equation 1 and the σ variance is included in equation

2 are used as shown in the next equation, equation 3 (Sellow et al.; 2002) where the probability density of framework difference is equal to the following:

$$p\ (FD) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\ \frac{(FD-\ \mu)^2}{2\sigma^2}\right) \ldots\ldots\ldots\ldots\ldots\ldots\ldots (3)$$

Where σ^2 is the variance of the frame difference and is equal to twice the camera noise variance.

3.3.4.2 Background removal:

When the frameworks differences are calculated, the pixels which related to the background area will have a value that nearly equal to zero, as mentioned previously the background is fixed. In several cases, some of the pixels that related to the background area may not close to zero. Throughout a comparison between any two differences, these values assigned to zero, assuming, *FDi* and *FDj*. Hence, the background area is removed and just the area of the moving object will include "nonzero pixels' values".

3.3.4.3 Background Registration:

A public tracking program is to get prominent areas from the provided video clip utilizing a technique of "learned background modeling". This includes eliminating every image from the view of the background and thresholding the output image from the difference to set foreground image. The fixed pixels are specified and processed to make the first registered image of background. Clear that the vehicle considered as a set of pixels move in a cohesive way, either as a darker background over a lighter area or vice versa. Usually, determining which the tracking object is hard because of the vehicle' color may be same as the background color, or some portion of it. This leads to wrong enumeration for the vehicles.

3.3.4.4 Detection for Foreground (Object Tracking):

Most views that based on the "traffic monitoring system" should be able to track vehicles via the sequence of the video. The process of tracking the objects assists in removing multiple counts in the applications of vehicle counting and it assists in eliciting helpful information while calculating speeds of the vehicle. To improve the vehicle kind and to correct faults that introduced because of obstructions, the tracking information can be utilized. To gain the "foreground dynamic objects", the background image is eliminated from the video frameworks after registering the fixed objects. Post processing is done on the "foreground dynamic objects" to decrement the noise' overlap.

3.3.4.5 Post Processing:

In several cases because of irregular motion of the object and the camera noise, constantly noise' areas are existed both in the background area and object area. Most of the techniques of the post processing are utilized to the acquired image after the removal process in the background. First of all, the arranged statistics filters are utilized, which are locative filters and its response is depend on ranking the pixels that include in the image region surrounded using the filter. The process of determining the filter response at any point will be made by means of ranking the result. The proposed algorithm utilizes Median filter, which considered as the best recognizable and arranged statistics filter. The filter substitutes the pixel value via the gray' levels' median in its neighboring pixels by using the equation (4) presented below:

$$\hat{f}(x, y) = median\{g(s, t)\} \quad \ldots\ldots(4)$$

where:

g(s,t) is image before using median filter

$\hat{f}(x, y)$ is the image after using the filter represented in binary

According to above; the output image is transformed into a binary image after using the "Median filter". Later, the technique of "morphological opening" is used on the binary image.

However, Object tuning is a post processing technique where in a number of applications is utilized. In the suggested algorithm, coplanar filter is applied for noise removing in both background and object. This technique of post processing is preferred to be applied on the "foreground image", due to the existence of non- straight object boundaries. At the end of object tuning phase a binary representation for the detected object' image is made, termed mask1.

3.3.4.6 Object Identification:

The acquired image after the pre-processing step has comparatively less noise, so that the background region is fully removed. Now, if the pixel values of the image are greater than a specific threshold then, the pixels substituted by the original framework' pixels. Hence, the moving object is determined via this process.

3.3.4.7 Object Counting:

In the counting process, its input image is the "tracked binary image" mask1. To detect the existence of an object using Quad tree Decomposition; the image scanned from top to bottom. One variable are preserved; ***count*** which track the number of vehicles and save count. However, this notion is performed for the full image, the final number of objects is currently in "variable count". Consequently, a- completely good accuracy of count is finished. Occasionally, because of obstructions, two objects are integrated together and dealt as one entity.

3.4 Summary

The suggested method introduces an effective algorithm for finding out a moving object utilizing the technique of background elimination. This chapter illustrates an algorithm that developed to pursue and count dynamic objects in an effective manner. The system of tracking depends on a group of temporal difference and interconnection matching. The system integrates simple knowledge domain effectively about classes with measures of domain statisticians to determine target objects in the existence of partial obstructions and fuzzy poses in which the vehicles are moving. The "background clutter" is rejected in an effective way.

The threshold taken is not fixed it can vary according to our perception. The use of threshold (Th) is just to separate the objects' pixels from the background. Finding the average intensity of the background frame, the threshold can be calculated as Th = 0.5.(average intensity). This threshold can be applied for background subtraction irrespective of the video and is pretty adaptive too.

However, Coplanar filter used in different practical applications, due to its performance of computing energy of the signal (image, text, voice,….), beside that coplanar filter differs from other filters by energy concept, since most of filters like smooth, edge and median filters are assumed to be time domain filters, while other filters like wavelet assumed to be frequency domain filters, but coplanar work in energy domain not time or frequency domains, so it assumed one of the best filters in binarization image, since it deals with impulsive noise removal, piecewise smoothing and sharp edge preservation.

Chapter Four
Results and Discussion

Chapter Four: Results and Discussion

4.1 Introduction

In this research, the traditional background subtraction algorithm is proposed. Due to this algorithm's limitation presented in chapter one, the researcher adopt an enhancement for this traditional algorithm based on applying both the Coplanar Filter and Quadtree Decomposition. After that, and according to the results obtained from the comparison between the traditional algorithm and the enhancement adoption, the last will be used in an applicable example. This example is an implementation of system designed for counting cars and tracing them using MATLAB, termed car tracking system. Counting the number of cars in imported videos will be the main use of this system. However, the aforementioned comparison between the proposed algorithm and the traditional version of this algorithm is then performed to estimate the performance of each one of them based on comparing the number of counted cars in each frame with the actual number of cars in that frame.

In the implemented system, a video that records the traffic in a specific street is initially uploaded to the system. The imported video is then segmented into a specific number of frames; images where each frame is then preprocessed and filtered. Each one of the preprocessed frames is then compared with an image for that traffic without cars using the enhanced background subtraction algorithm. This is performed based on comparing each pixel in the processed frame with its corresponding pixel in the street image without cars. When the difference among pixels is larger than a certain threshold equal to 0.5 is taken. the pixel in the processed frame is considered as a foreground one and then it is represented in white in the output image. On the other hand, when the difference among pixels is smaller than that threshold, the pixel is considered as a background one and then it is represented in black in the output image. At the end of this process, a black and white image is obtained, where the white objects represent the cars in that traffic, while the black ones stand for the background. After that, the number of white objects in the final image is counted. The Figure 8 shows the stages summary of the implemented system.

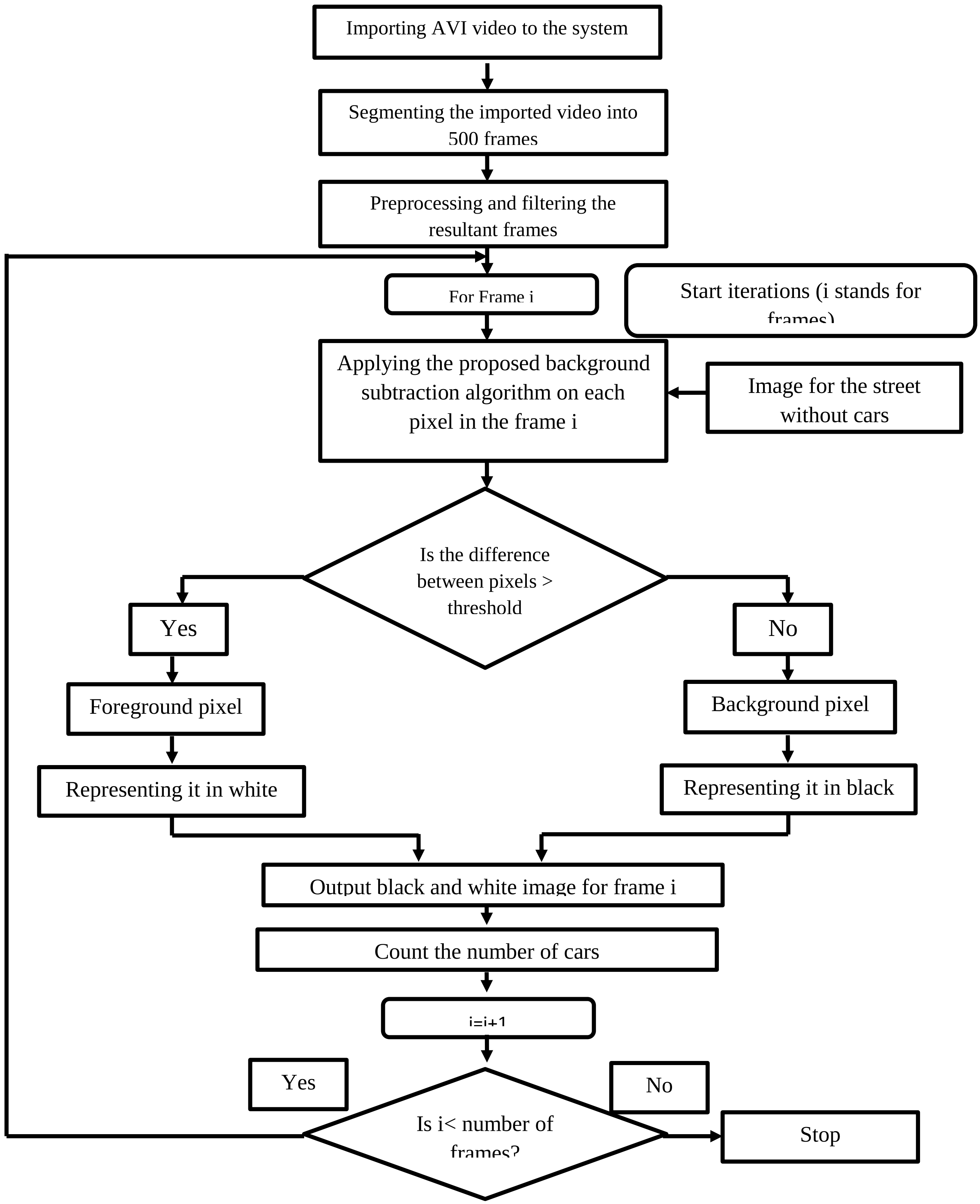


Figure 8 flowchart of proposed background subtraction

4.2 Importing and Segmentation

To evaluate the implemented car tracking system, a video record for the traffic in one side of a street is imported to the system. This video which we have taken is with period equal to 33s segmented into 500 frames (images) resulting 15 frame per second .the Figures 9,10,11,12 show some of these frames.

Figure 9 Frame example

Figure 10 Frame example

Figure 11 Frame example

Figure 12 Frame example

4.3 Preprocessing of Frames

After segmenting the input video into frames, each one of the resultant frames is then preprocessed. In the preprocessing stage, each frame is initially resized into 650X400 pixels. After that, it is filtered to remove noises based on applying morphological filter using the ***strel*** command, which create structuring square frames with dimension equal to four pixels in each side. Conventional methods of eliminating regions with noise are made within the use of the morphological procedures in order to filter the smaller regions out. The one operation which is valuable for removing the background noise termed close operation while the other one in which removing noise within the object region is efficiently performed is termed the open operation. However, the areas of noise regions which are bigger than the structuring element cannot be eliminated throughout the use of either the close or open operations. For eliminating noise regions whose area is big, bigger structuring element must be performed. The aforementioned will not results in increasing the complexity of computation only, nevertheless it will results in disgracing the object boundary accuracy as well. (Chen et al.; 2002).

4.4 Application of Traditional and Enhanced Background Subtraction Algorithm

Both the traditional (blob analysis) and enhanced (Coplanar and Quad tree Decomposition) background subtraction algorithm are applied on the imported video to show the degree of enhancement that achieved after applying the enhanced algorithm.

For the traditional background subtraction algorithm, the foreground pixels in each frame are detected. After that, the morphological opening is applied to remove noises in the foreground using the command ***imopen***, which carries out a morphological opening on the detected foreground with the current structuring frame. The morphological opening process is an erosion that followed by a dilation in the same frame. The connected components with the specified minimum area which represent the cars are detected and counted using the ***size*** command. These processes are performed based on applying the following commands.

```
% Blob Analysis
% Detect the foreground in the current video frame
 foreground = step(foregroundDetector, frame);
% Use morphological opening to remove noise in the foreground
 filteredForeground = imopen(foreground, se);
 bbox = step(blobAnalysis, filteredForeground);
% Detect the connected components with the specified minimum area
 numCars = size(bbox, 1);
```

On the other hand, for the enhanced background subtraction algorithm, the coplanar and Quadtree decomposition is initially applied on each frame with the median filter using the command ***medfilt2***, which is a nonlinear operation that decreases the salt and pepper noise. This command performs a median filtering on the foreground, where each output pixel includes the median value in 20X20 neighborhood pixels around the related pixel in the frame.

After that, noises are removed using the coplanar filter. With this filter, the ***bwboundaries*** command is used to trace the external boundaries of objects in the foreground and boundaries of holes within its objects and the ***qtdecomp*** command is applied to segment the square image into four square blocks with the same size, where each block is then tested to decide if it meets specific homogeneity criterion or not. When the block meets that criterion, it is not segmented again. Else, it is sub-segmented into other four blocks, where the test is then applied on each block and so on. This procedure is repeated until each one of the blocks meet the specified criterion. Due to the sub-divisions, results may have blocks with different sizes. Therefore, ***qtdecomp*** does not generate blocks smaller or bigger than 265 pixels in size, where blocks larger than 265 pixels are split regardless if they meet the threshold condition or not. Moreover, qtdecomp (Quadtree decomposition) divides a square image into four equal-sized square blocks, and then tests each block to see if meets some criterion of homogeneity. If a block meets the criterion, it is not divided any further. If it does not meet the criterion, it is subdivided again into four blocks, and the test criterion is applied to those blocks. This process is repeated iteratively
until each block meets the criterion. The result may have blocks of several different sizes.

The ***repmat*** command is then applied to generate a blocks where its size is ***uint8(0)*** by the size of the Quadtree decomposition. After that, the ***qtsetblk*** command is applied to set block values in quadtree decomposition, where it replaces each dim-by-dim block in the resultant quadtree decomposition of blocks with the related dim-by-dim block in values array. This command depends on S, which is the returned sparse matrix by the command ***qtdecomp***, where it includes the quadtree structure. After completing the iteration process, the number of cars in each frame is counted using the ***length*** command, which finds the number of elements in the traced region boundaries in that frame. The following commands summarize the process of the enhanced algorithm.

```
%% Coplanar+Quadtree Decomposition
 %Removing Noise + coplanar.
    filterforeground = medfilt2(foreground,[20 20]);
    B = bwboundaries(~filterforeground);
    S = qtdecomp(imresize(filterforeground, [256 256]),  0.1);


    blocks = repmat(uint8(0),size(S));
    for dim = [512 256 128 64 32 16 8 4 2 1];
      numblocks = length(find(S==dim));
      if (numblocks > 0)
        values = repmat(uint8(1),[dim dim numblocks]);
        values(2:dim,2:dim,:) = 0;
        blocks = qtsetblk(blocks,S,dim,values);


      end
    end
    blocks(end,1:end) = 1;
    blocks(1:end,end) = 1;
    countCars = length(B);
```

4.5 Offline Evaluation

In the offline evaluation, images are imported to the implemented system to be evaluated. A background image and a test one are initially inserted into the system as shown in Figure 13.

Figure 13 Background and test images

Both images are then converted into grayscale images. The resultant grayscale background image as shown Figure 14.

Figure 14 Grayscale background image

The resultant grayscale test image and the result of applying the Quadtree decomposition to detect cars are shown below. As shown in Figure 15 the number of cars are 3 and the results of Quadtree decomposition gives output 3 also.

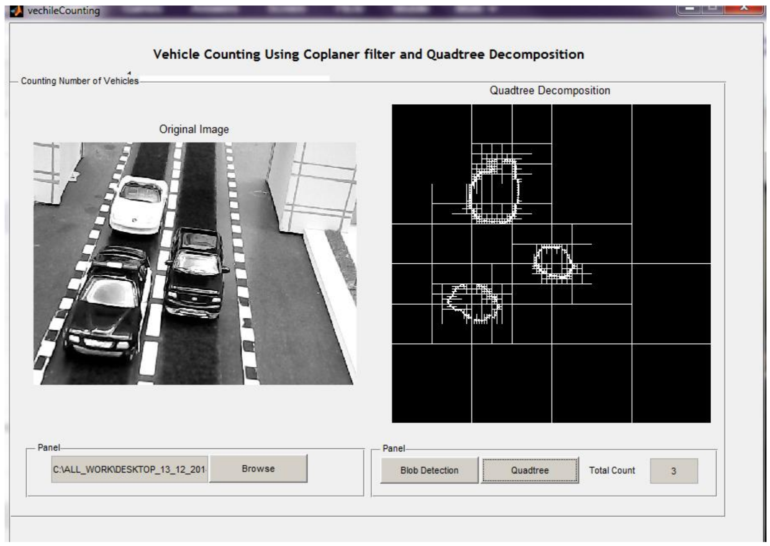


Figure 15 grayscale test image and the result of applying the Quadtree decomposition

After that, the difference among both images is computed as explored earlier by traditional background subtraction, see Figure 16 which shows the resultant black (background) and white (foreground) image. This figure also shows the representation of the detected foreground objects (cars) on the original test image. According to Figure 16, the counted number of cars is 4, while the actual number of cars is 3.

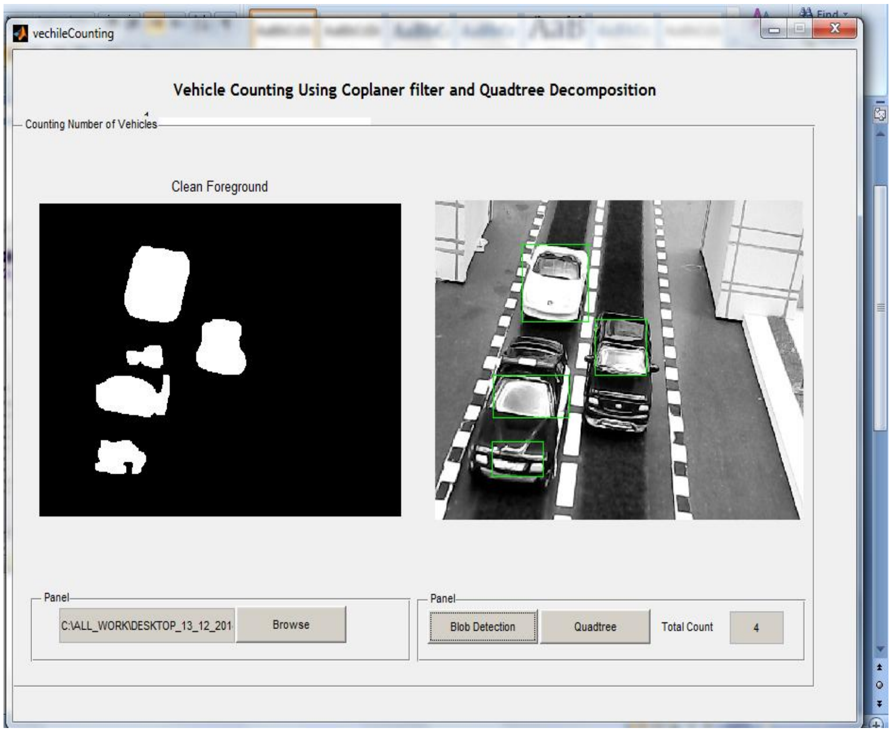


Figure 16 Black and white output images

Another evaluation for the system is conducted based on inserting another test image that has more cars as shown in Figure 17.

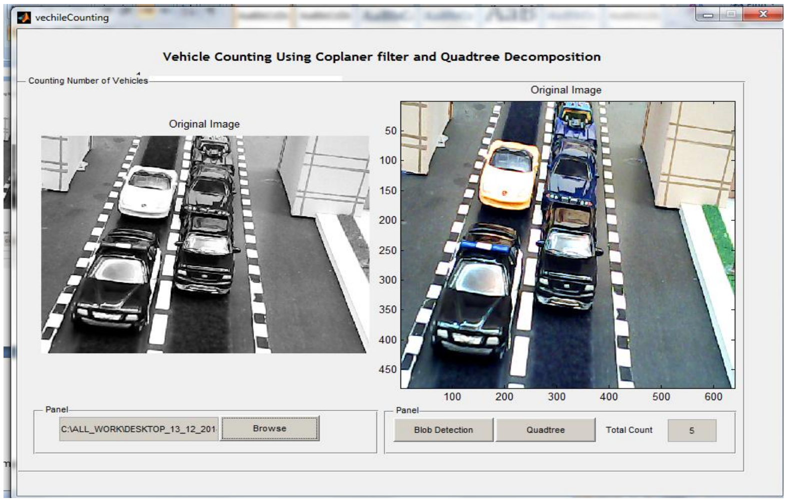


Figure 17 colored and grayscale test image

The application of the Quadtree decomposition on the grayscale test image to detect cars is shown in the following figure. As shown in Figure 18, the counted number of cars is 5, while the actual number of cars is 5.

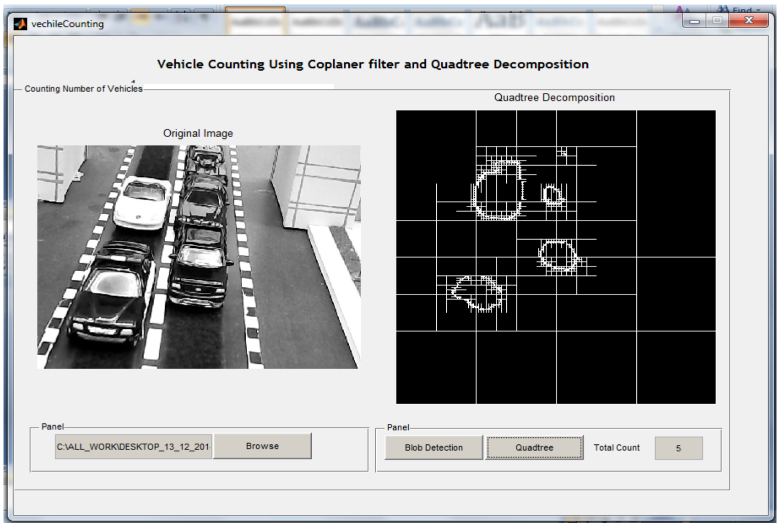


Figure 18 application of the Quadtree decomposition

The Figure 19 shows the representation of the detected foreground objects (cars) on the test image, the counted number of cars is 4, while the actual number of cars is 5.

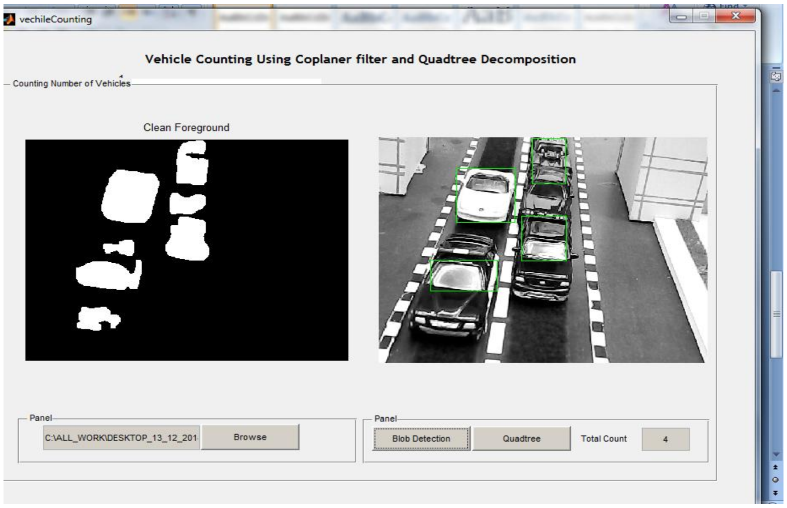
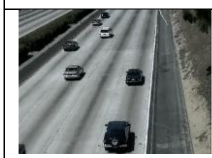
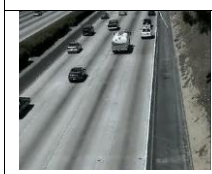
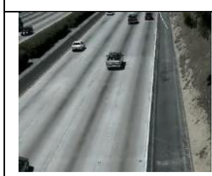


Figure 19 Output image

4.6 Online Evaluation

As proposed earlier, a recorded video is imported to the implemented system in both cases; traditional and enhanced background subtraction algorithm. The video is segmented into frames where each frame is then compared with a defined background image in both cases. The system in both cases counts the number of cars in the street. These counted numbers in each case are then compared with the actual number of cars for each frame to determine the optimal background subtraction algorithm. An example for some of the obtained results are shown in Table 1.

Table 1: An example for somme of the obtained results for both the traditional and enhanced algorithms.

Image number (Real image)	No. of counted cars by the traditional algorithm	No. of counted cars by the enhanced algorithm	Real no. of cars	Error in the traditional algorithm	Error in the enhanced algorithm	Accuracy of the traditional algorithm	Accuracy of the enhanced algorithm
	4	6	7	3	1	57.14%	85.71%
	3	5	6	3	1	50%	83.33%
	4	5	5	1	0	80%	100%
	3	4	4	1	0	75%	100%
	4	5	6	2	1	66.66%	83.33%

A comparison among both algorithms based on the number of cars in frames is demonstrated in the following subsections below.

- **Detection of one car**

As shown in the table 2, the traditional algorithm fails in detecting one car, while the enhanced algorithm detects the car in both sides and provides 100% detection accuracy.

Table 2 Results of detecting one car

Frame	No. of counted cars by the traditional algorithm	No. of counted cars by the enhanced algorithm	Real no. of cars	Error in the traditional algorithm	Error in the enhanced algorithm	Accuracy of the traditional algorithm	Accuracy of the enhanced algorithm
	0	1	1	1	0	0%	100%
	0	1	1	1	0	0%	100%
	0	1	1	1	0	0%	100%

- **Detection of two cars**

As shown in the table 3, the traditional algorithm average accuracy in detecting two cars is 25% , while the enhanced one detects the car in both sides offers 75% detection accuracy.

Table 3 Results of detecting two cars

Frame	No. of counted cars by the traditional algorithm	No. of counted cars by the enhanced algorithm	Real no. of cars	Error in the traditional algorithm	Error in the enhanced algorithm	Accuracy of the traditional algorithm	Accuracy of the enhanced algorithm
	1	2	2	1	0	50%	100%
	1	2	2	1	0	50%	100%
	0	1	2	2	1	0%	50%
	0	1	2	2	1	0%	50%

- **Detection of three cars**

As shown in the table 4, the traditional algorithm average accuracy in detecting three cars is 58.3 % , while the enhanced one detects the car in both sides offers 100% detection accuracy.

Table 4 Results of detecting three cars

Frame	No. of counted cars by the traditional algorithm	No. of counted cars by the enhanced algorithm	Real no. of cars	Error in the traditional algorithm	Error in the enhanced algorithm	Accuracy of the traditional algorithm	Accuracy of the enhanced algorithm
	2	3	3	1	0	66.66%	100%
	1	3	3	2	0	33.33%	100%
	2	3	3	1	0	66.66%	100%
	2	3	3	1	0	66.66%	100%

- **Detection of four cars**

As shown in the table 5, the traditional algorithm average accuracy in detecting four cars is 60.4% , while the enhanced one detects the car in both sides offers 93.75% detection accuracy.

Table 5 Results of detecting four cars.

Frame	No. of counted cars by the traditional algorithm	No. of counted cars by the enhanced algorithm	Real no. of cars	Error in the traditional algorithm	Error in the enhanced algorithm	Accuracy of the traditional algorithm	Accuracy of the enhanced algorithm
	2	3	4	2	1	50%	75%
	5	4	4	-1	0	66.66%	100%
	2	4	4	2	0	50%	100%
	3	4	4	1	0	75%	100%

- **Detection of five cars**

As shown in the table 6, the traditional algorithm average accuracy in detecting five cars is 55% , while the enhanced one detects the car in both sides offers 95% detection accuracy.

Table 6 Results of detecting five cars

Frame	No. of counted cars by the traditional algorithm	No. of counted cars by the enhanced algorithm	Real no. of cars	Error in the traditional algorithm	Error in the enhanced algorithm	Accuracy of the traditional algorithm	Accuracy of the enhanced algorithm
	3	5	5	2	0	60%	100%
	4	5	5	1	0	80%	100%
	1	4	5	4	1	20%	80%
	3	5	5	2	0	60%	100%

- **Detection of six cars**

As shown in the table 7, the traditional algorithm average accuracy in detecting six cars is 62.49% , while the enhanced one detects the car in both sides offers 91.66% detection accuracy.

Table 7 Results of detecting six cars

Frame	No. of counted cars by the traditional algorithm	No. of counted cars by the enhanced algorithm	Real no. of cars	Error in the traditional algorithm	Error in the enhanced algorithm	Accuracy of the traditional algorithm	Accuracy of the enhanced algorithm
	3	5	6	3	1	50%	83.33%
	3	5	6	3	1	50%	83.33%
	4	6	6	2	0	66.66%	100%
	5	6	6	1	0	83.33%	100%

- **Detection of seven cars**

As shown in the table 8, the traditional algorithm average accuracy in detecting seven cars is 53.56% , while the enhanced one detects the car in both sides offers 82.13% detection accuracy.

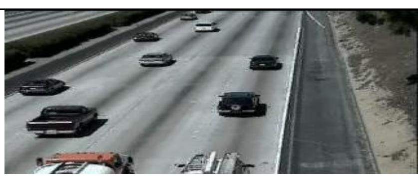
Table 8 Results of detecting seven cars

Frame	No. of counted cars by the traditional algorithm	No. of counted cars by the enhanced algorithm	Real no. of cars	Error in the traditional algorithm	Error in the enhanced algorithm	Accuracy of the traditional algorithm	Accuracy of the enhanced algorithm
	4	6	7	3	1	57.14%	85.71%
	3	5	7	4	2	42.85%	71.42%
	4	6	7	3	1	57.14%	85.71%
	4	6	7	3	1	57.14%	85.71%

- **Detection of eight cars**

As shown in the table 9, the traditional algorithm average accuracy in detecting eight cars is 71.87% , while the enhanced one detects the car in both sides offers 87.5% detection accuracy.

Table 9 Results of detecting eight cars

Frame	No. of counted cars by the traditional algorithm	No. of counted cars by the enhanced algorithm	Real no. of cars	Error in the traditional algorithm	Error in the enhanced algorithm	Accuracy of the traditional algorithm	Accuracy of the enhanced algorithm
	6	7	8	2	1	75%	87.5%
	7	8	8	1	0	87.5%	100%
	5	6	8	3	2	62.5%	75%
	5	7	8	3	1	62.5%	87.5%

- **Detection of nine cars**

As shown in the table 10, the traditional algorithm average accuracy in detecting nine cars is 69.43% , while the enhanced one detects the car in both sides offers 83.32% detection accuracy.

Table 10 Results of detecting nine cars

Frame	No. of counted cars by the traditional algorithm	No. of counted cars by the enhanced algorithm	Real no. of cars	Error in the traditional algorithm	Error in the enhanced algorithm	Accuracy of the traditional algorithm	Accuracy of the enhanced algorithm
	6	9	9	3	0	66.66%	100%
	6	7	9	3	2	66.66%	77.77%
	6	7	9	3	2	66.66%	77.77%
	7	7	9	2	2	77.77%	77.77%

Regarding the aforementioned tables, the enhanced background subtraction algorithm outperforms the traditional algorithm in term of counting cars number in the frame. The average accuracy of the traditional algorithm for all frames is 47.01%, while it is 81.19% for the enhanced algorithm, and both of them have the same running time on the same computer.

Chapter Five:
Conclusion and Future Works

Chapter Five: Conclusion and Future Works

5.1 Conclusion

This research provides an enhancement adoption for the traditional background subtraction algorithm based on using both the Coplanar Filter; to improve the detection of all edge pixels, and Quadtree Decomposition; to divide images into blocks which are more homogenous than the original image. However, Both the enhanced and traditional background subtraction algorithms are then applied to implement a car tracking system with the use of MATLAB to count the number of cars in a specific street. Both performance and accuracy of these algorithms are evaluated according comparing the counted number of cars by each algorithm with the actual number of cars in that street in different times.

Two evaluation stages were executed; offline and online stages are applied to assess the performance of the implemented car tracking system using MATLAB. In the offline stage, images for the traffic in a certain street with an image for that street without cars are initially imported to the implemented system. These images are then converted into grayscale format where the Quadtree decomposition is then applied on images to compare between them. The comparison then results in a white (foreground) and black (background) image, where the white regions represent cars. The detected car regions are then represented on the original image where then the number of these regions (cars) is counted.

In the online stage, a video recorded for the traffic in a specific street is initially imported to the implemented system. The imported video is then segmented into 500 frames (images). After that, both the traditional (blob analysis) and enhanced (Coplanar and Quad tree Decomposition) background subtraction algorithms are applied on the resultant frames to detect and count the number of cars in each frame. In both algorithms, each one of these frames is then preprocessed based on resizing it into 650X400 pixels and filtering the resized frame to remove noises based on applying a morphological filter. The traditional background subtraction algorithm depends on comparing each pixel in the background image with its corresponding pixel in each frame, where when the difference among them is larger than a certain threshold, that pixel is considered as a foreground (white) one in the output image. Else, it is considered as a background (black) pixel in the output image. In the enhanced background subtraction algorithm, the coplanar filter is applied on each frame to remove noises, enhance edges and trace the external boundaries of objects and holes within objects. In addition, the Quadtree decomposition is applied to segment each frame into four square blocks with the same size to be tested and segmented into sub-blocks till each one of the blocks meet a specified criterion. After applying both algorithms, the number of cars in each frame is counted. Function in MATLAB blob analysis sees the objects and the function size counts it . while the other uses segmentation .

The frames of the imported video include various numbers of cars. These counted numbers by each algorithm are compared with the actual number of cars for each frame to determine the optimal background subtraction algorithm. Results demonstrate that the enhanced background subtraction algorithm outperforms the traditional one in the accuracy of counting the number of cars in all frames. The resultant average accuracy of the traditional background subtraction algorithm is 47.01%, while it is 81.19% for the enhanced one.

5.2 Future Works

A coplanar Filter and Quad tree Decomposition dependent cars tracking system is introduced and implemented in this work. Results prove the accuracy of this system in detecting and counting cars in a specific street. However, this work can be enhanced in the future based on performing the following:

- Applying the implemented system on videos recorded for cars in two-side streets.
- Importing videos for cars with various sizes.
- Enhancing the system to detect and count cars and other objects in the same time.
- Applying other preprocessing stages for more enhancements of images.
- Enhancing the system to detect and track the motions of objects in videos.

References

- Anchez-Ferreira, C. S. Mori, J. Y., and Llanos, C. H. (2012). Background Subtraction Algorithm for Moving Object Detection on FPGA.
- Anitha, A. Gayatri, J. and Ashwini, K. (2013). Automatic recognition of object detection using Matlab. ***International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)***. Vol. 2.
- Augustin, M. Juliet, S. and Palanikumar, S. (2011). Motion and Feature Based Person Tracing In Surveillance Videos. ***IEEE.***
- Cheung, S. C. and Kamath, C. (2015). Robust Background Subtraction with Foreground Validation for Urban Traffic Video. ***Lawrence Livermore National Laboratory.***
- Cucchiara, R. Piccardi, M. and Prati, A. (2003). Detecting moving objects, ghosts, and shadows in video streams. ***IEEE Transactions on Pattern Analysis and Machine Intelligence.*** **25**, 1337-1342.
- Fan, L. Tan, C. L. and Fan, L. (2001). Binarizing Document Image using Coplanar prefilter. ***National University of Singapore.*** 34-38.
- Hwang, P. Eom, k. Jung, J., and Kim, M. (2009).A statistical approach to robust background subtraction for urban traffic video. ***Second International Workshop on Computer Science and Engineering.*** 177-181.
- Lee, D. S. Hull, J. and Erol, B. (2003). A Bayesian framework for Gaussian mixture background modeling. ***in Proceedings of IEEE International conference on Image Processing,*** Barcelona, Spain.
- Mahamuni, P. D, Patil, R. P. and Thakar, H.S. (2014). Moving object detection using background subtraction algorithm using Simulink. ***IJRET: International Journal of Research in Engineering and Technology.*** Vol.03. [online] available at: http://www.ijret.org.
- Moeslund, B., and Hiton, A. (2006). A survey of advances in vision-based human motion capture and analysis. ***Computer Vision and Image Understanding***. 90–126.
- Nagendran, A. Dheivasenathipathy, N. Nair, and R. Sharma, V. (2014). Recognition and tracking moving objects using moving camera in complex scenes. ***International Journal of Computer Science, Engineering and Applications (IJCSEA).*** Vol.4, No.2.
- Power, P. W. Schoonees, J. A. (2002). Understanding background mixture models for foreground segmentation. ***In Proceedings Image and Vision Computing New Zealand,*** 267-271.

- Raghtate, G. Tiwari, A. (2014). Moving Object Counting in Video Signals. ***International Journal of Engineering Research and General Science.*** 415-420.
- Rostamianfar, O. Janabi-Sharifi, F. and Hassanzadeh, I. (2006). Visual Tracking System for Dense Traffic Intersections. ***Canadian Conference on Electrical and Computer Engineering.*** 2000-2004.
- Sajjad, K. M. (2010). Automatic License Plate Recognition using Python and OpenCV. ***M.E.S. College of Engineering, Kuttippuram, Kerala.***
- Samuel, G. and LIU, H. (2010). Comparison of complex-background subtraction algorithms using a fixed camera***. Intelligent System and Robotics University of Portsmouth***. 1-4.
- Sigari, M. H. Mozayani N. and Pourreza H. R. (2008). Fuzzy Running Average and Fuzzy Background Subtraction. ***International Journal of Computer Science and Network Security***, 138-143.
- Tamersoy, B. (2009). Background Subtraction. ***The University of Texas at Austin.***
- Tumul, A. and Nagalaxmi, T. (2014). Object Identification Based on Background Subtraction and Morphological Process. ***International Journal of Advanced Research in Computer and Communication E Engineering***. Vol. 3.
- Vibha, L. Venkatesha, M. Rao, P. G. Suhas, N. Shenoy, P. D. Venugopal, K. R and Patnaik, L. M. (2008). Moving Vehicle Identification using Background Registration Technique for Traffic Surveillance. ***Proceedings of the international multi-conference of engineers and computer scientists.***

Appendix: Code Listing

Author name: Fayez Kamal Alzaghal

Program Name: MATLAB

Date: 15/10/2015

Matlab Version: 2014a

Computer processor: Intel ® I7

Operating system: win 8.1 pro 64-bit

Code Offline Evaluation:

```
function varargout = vechileCounting (varargin)
% VECHILECOUNTING MATLAB code for vechileCounting.fig
% VECHILECOUNTING, by itself, creates a new VECHILECOUNTING or raises the existing singleton*.

% H = VECHILECOUNTING returns the handle to a new VECHILECOUNTING or the handle to the existing singleton*.

gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @vechileCounting_OpeningFcn, ...
                   'gui_OutputFcn',  @vechileCounting_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
function vechileCounting_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
I = imread('background.jpg');                   %Read Background image
axes(handles.axes2);                            %Control Axes.
image(I);                                       %Show in the axes
title(handles.axes2, 'Background Image');
% Update handles structure
guidata(hObject, handles);

% UIWAIT makes vechileCounting wait for user response (see UIRESUME)
% uiwait(handles.figure1);


% Outputs from this function are returned to the command line.
function varargout = vechileCounting_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args % hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data
% Get default command line output from handles structure
```

```
varargout{1} = handles.output;


% Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data




function edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit1
% eventdata  reserved - to be defined in a future version of MATLAB


function edit1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit1
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data

[filename filepath] = uigetfile({'*.jpg'; '*.png'; '*.*'}, 'File Selector');      %Browse the image

x=strcat(filepath,filename);                          %combine the path and the filename
set(handles.edit2,'String',x);                        %display it on the edit box;
ImageRead = imread(x);
axes(handles.axes3);
image(ImageRead);
title(handles.axes3, 'Original Image');

function edit2_Callback(hObject, eventdata, handles)
% hObject    handle to edit2
% handles    structure with handles and user data
```

```
% str2double(get(hObject,'String')) returns contents of edit2 as a double


%Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit2

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton4
% eventdata  reserved - to be defined in a future version of MATLAB


global bbox;
I1=rgb2gray(imread('background.jpg'));                    %Reading Background image
filename=get(handles.edit2,'String');                 %Getting File name from Edit Box
I2=rgb2gray(imread(filename));                     %Reading Original Image for Counting
Vehicle
I3=imsubtract(I2,I1);                              %Background Substraction.
I4=im2bw(I3,0.1);                                  %Converting to Black and White.
%Threshold = 0.1;
foreground=medfilt2(I4,[3020]);                       %Removing Noise.

title(handles.axes2, 'Binary Image');

se = strel('square', 10);
filteredForeground = imopen(foreground, se);
axes(handles.axes2)
imshow(filteredForeground);

title('Clean Foreground');
blobAnalysis = vision.BlobAnalysis('BoundingBoxOutputPort', true, ...
    'AreaOutputPort', false, 'CentroidOutputPort', false, ...
    'MinimumBlobArea', 150);
bbox = step(blobAnalysis, filteredForeground);
%result = insertShape(I2, 'Rectangle', bbox, 'Color', 'green');

axes(handles.axes3);
imshow(I2);
numCars = 0;
```

```
for i = 1 : size(bbox, 1)
    Area(i) = bbox(i, 3) * bbox(i, 4);
    if Area(i) > 4500
        Area(i)
        numCars = numCars + 1;
        rectangle('Position', bbox(i, :), 'EdgeColor', 'g');
    end
end
numCars = set(handles.edit3, 'String', num2str(numCars));
title('Detected Cars');


% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton5
% eventdata  reserved - to be defined in a future version of MATLAB

global foreground;
%Quadtree Decomposition.
I1=rgb2gray(imread('background.jpg'));
 %Reading Background image
filename=get(handles.edit2,'String');
 %Getting File name from Edit Box
I2=rgb2gray(imread(filename));
 %Reading Original Image for Counting Vehicle
I3=imresize(imsubtract(I2,I1),[512512]);
 %Background Substraction.
I4=im2bw(I3);
%Converting to Black and White.
%Threshold = 0.1;
foreground=medfilt2(I4,[4030]);
 %Removing Noise.
B = bwboundaries(~foreground);
axes(handles.axes2);
hold(handles.axes2, 'on');
imshow(I2);
title('Original Image');


%% Quadtree decomposition.
S = qtdecomp(foreground,  0.1);
blocks = repmat(uint8(0),size(S));
for dim = [512 256 128 64 32 16 8 4 2 1];
  numblocks = length(find(S==dim));
  if (numblocks > 0)
```

```
    values = repmat(uint8(1),[dim dim numblocks]);
    values(2:dim,2:dim,:) = 0;
    blocks = qtsetblk(blocks,S,dim,values);
  end
end

blocks(end,1:end) = 1;
blocks(1:end,end) = 1;
axes(handles.axes3)

imshow(blocks, []);
title(handles.axes3, 'Quadtree Decomposition');
set(handles.edit3, 'String', num2str(length(B) - 1));
function edit3_Callback(hObject, eventdata, handles)
% hObject    handle to edit3
% handles    structure with handles and user data


function edit3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit3

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

Code Online Evaluation:

```
clc
clear all
close all


fileFormat = {'Image', 'Count(Blob Analysis)' 'Count (Coplanar_Quadtree Decomposition)'};
x = dir(fullfile(pwd, '*.avi'));
fprintf('Choose a video file:\n');
for i = 1 : length(x)
   fprintf('%d. %s\n', i, x(i).name);
end
n = input('Enter a video file:');
filename = strcat('video', num2str(n), '.avi');
foregroundDetector = vision.ForegroundDetector('NumGaussians', 3, ...
   'NumTrainingFrames', 100);
blobAnalysis = vision.BlobAnalysis('BoundingBoxOutputPort', true, ...
   'AreaOutputPort', false, 'CentroidOutputPort', false, ...
   'MinimumBlobArea', 150);
videoReader = vision.VideoFileReader(filename);
videoPlayer = vision.VideoPlayer('Name', 'Detected Cars');
se = strel('square', 3);
count = 0;
while ~isDone(videoReader)
   count = count + 1;
   frame = step(videoReader);
   %read the next video frame
   imageFile =  strcat('image', num2str(count), '.jpg');
   imwrite(frame, imageFile);
   %% Blob Analysis
   % Detect the foreground in the current video frame
   foreground = step(foregroundDetector, frame);
    % Use morphological opening to remove noise in the foreground
   filteredForeground = imopen(foreground, se);
   bbox = step(blobAnalysis, filteredForeground);
   % Detect the connected components with the specified minimum area, and
   numCars = size(bbox, 1);
   %% Coplanar+Quadtree Decomposition
   filterforeground = medfilt2(foreground,[20 20]);
    %Removing Noise + coplanar.
   B = bwboundaries(~filterforeground);
   S = qtdecomp(imresize(filterforeground, [256 256]),  0.1);
```

```
% % % qtdecomp Quadtree decomposition.
% % % qtdecomp divides a square image into four equal-sized square blocks, and
% % % then tests each block to see if meets some criterion of homogeneity. If a
% % % block meets the criterion, it is not divided any further. If it does not
% % %  meet the criterion, it is subdivided again into four blocks, and the test
% % % criterion is applied to those blocks. This process is repeated iteratively
% % % until each block meets the criterion. The result may have blocks of
% % %  several different sizes.

    BB = repmat(uint8(0),size(S));
    for all_dimension = [512 256 128 64 32 16 8 4 2 1];
      numblocks = length(find(S==all_dimension));
      if (numblocks > 0)
        values = repmat(uint8(1),[all_dimension all_dimension numblocks]);
        values(2:all_dimension,2:all_dimension,:) = 0;
        BB = qtsetblk(BB,S,all_dimension,values);

% % % qtsetblk Set block values in quadtree decomposition.
% % % J = qtsetblk(I,S,DIM,VALS) replaces each DIM-by-DIM block in the quadtree
% % % decomposition of I with the corresponding DIM-by-DIM block in VALS. S is
% % % the sparse matrix returned by QTDECOMP; it contains the quadtree
% % %     structure. VALS is a DIM-by-DIM-by-K array, where K is the number of
% % %     DIM-by-DIM blocks in the quadtree decomposition.
% % %

      end
    end
    BB(end,1:end) = 1;
    BB(1:end,end) = 1;
    countCars = length(B);

    fileFormat = vertcat(fileFormat, {imageFile, numCars, countCars});

end
release(videoReader);
 % close the video file
xlswrite('vehicle.xlsx', fileFormat);
```