Estimating vehicle speed using image processing

Ammar Awni Abbass

University of Baghdad

Abstract:

This research uses image processing to estimate car speed. Given a sequence of real-time video of traffic images. The images are converted to monochrome images then edges are extracted and then quantified the resulting images to classify the objects and find the cars, after detecting the cars, they can be tracked in different frames and their speed can be estimated by calculating the position of the car object in different frames and compare it to a ground truth. This research uses a different approach to estimate speed, rather than using a reference object to estimate speed, the Matlab function is used and different relationships between the objects that are extracted from the image itself. This method does not need camera calibration and does not need any extra equipments other than a video camera attached to a computer and can be used by traffic police to monitor highways car or normal roads, and can be combined with a system to detect the license plate number to create a complete system to issue an automatic speed ticket for drivers who exceeds the speed limits. The system is tested in different roads and the results show the validity of the used method.

1 Introduction

Image processing has been widely applied to traffic analysis in the last decade for a variety of purposes. The field of traffic research is very wide and it has many goals that include queue detection, incident detection, vehicle classification, and vehicle counting, One of the most popular of these purposes is to attempt to estimate the speed of a moving car, a vehicle system measures the vehicle speed, and combined with a system to capture and recognize the image containing the license plate number of the over speeding vehicles, it can issue an automatic speeding ticket. In this research, we address the problem of estimating the vehicle speed using a very simple hardware (a laptop computer with a camera that does not need to be calibrated) and a very concise and effective software written using Matlab programming language is used. A new approach is presented for extracting vehicular speed information, given a sequence of real-time traffic images that is taken from a laptop camera that is put in the side of a two sided road to capture the images of that road when the vehicle pass through as shown in figure (1), the images are then processed by the program.

Figure1: A typical scene of the proposed method.

The structure of this paper presents the related work in literature in section 2, followed by the basic idea for the problem in section 3, the system layout (system used to solve this problem) is in section 4, and the implementation of the system with a breakdown and description of each of the elements in that system is in sections 5, the results are provided showing the effectiveness of the proposed method in section 6, the discussions of the results in sections 7, the graphical user interface is in section 8, the conclusions are found in section 9, the future work is found at the end of the document in section 10.

1.1 Problem Definition

The task here is to automatically estimate vehicle speed from video sequences, acquired from a laptop camera from a road side. Assuming that the studied road segments are planar and straight. Vehicles are automatically detected and tracked along frames. No reference object is placed in the street to guide the program in the process of detecting the vehicles or their speed.

In this work, a 160x120 indexed images at a frame rate of 29 frames per second (fps) were used. These are demonstrated to be adequate for reliable analysis, as well as being small enough to allow efficient processing.

1.2 Motivation

As traffic speed monitoring and vehicle counting using buried induction loops involves relatively high installation and maintenance costs, highway authorities switch to noninvasive technologies such as microwave radar, infrared, ultrasound,

And video detection. The advantages of non-optical techniques are their relative independence from weather and light conditions, however, most of these devices need to be installed centered above the single lane they service. The task here is to find a cost effective method to detect the vehicles' speed that is cost effective with moderate error rate and low response time.

1.3 Limitations

Advanced video detection is capable of monitoring several lanes simultaneously from a side mount position and delivers an image of the traffic situation, but performance suffers from poor lighting or bad weather conditions. Furthermore, Real-time video detection is computationally expensive and relies on high-performance signal processing hardware, large amounts of memory, and high-bandwidth data links.

The density of installation along a highway route is limited and may not reach the density necessary to acquire enough data for reliable traffic flow prediction.

2 Related Work

The literature on estimating vehicles' speed falls in several categories. Many papers analyze low frame-rate video taken from an un-calibrated camera and use it to estimate mean traffic speed (3), using this approach takes advantage of a known relationship between traffic speed and traffic density to make tracking of individual vehicles unnecessary additionally, the traffic queue length measurement using an Image Processing Sensor has occupied a great space in the related work, they have developed new measurement algorithms and a new hardware, and they have succeeded in the development of a new image sensor (4).

Two references is particularly relevant to our work. The papers (1,2) discusses the development of an unconventional, yet an effective method for estimating vehicle speed, it presents an algorithm that allows automatic 1D measurements and subsequent estimation of vehicle speed form single un-calibrated images. It requires minimal external data (one known distance), if basic assumptions are adopted.

3 Basic ideas

3.1 Vehicle Tracking:

After recording the video from the camera, the video image sequence is stored in the Matlab workspace; two simple techniques are used for tracking the objects (vehicles): frame differencing and background subtraction. The functions in the Image Processing Toolbox are used. The absolute difference between successive frames can be used to divide an image frame into changed and unchanged regions. Since only the vehicle moves, the changed region is expected to be associated only with the vehicle, or possibly with its shadow.

To begin, each frame is converted to an indexed image, then to a monochrome image, the unique size of the vehicle and the fact that the vehicle moves and the background doesn't move enable efficient vehicle tracking.

3.2 Background Subtraction

Another approach to tracking the vehicle is to estimate the background image and subtract it from each frame. The approach here is to find the pixel-wise maximum among several neighboring frames.

That's exactly what morphological dilation does, if a structuring element oriented is used along the frame dimension. Next, the absolute difference between each frame and its corresponding background estimate is computed.

4 System Layout

The system model can be broken into three primary stages:

1- Preprocessing.

2- Data Reduction.

3- Feature analysis.

In preprocessing, the operation is performed to make data reduction and analysis easier, the video stream is converted to an indexed image then to a binary image. The binary (Black and White) image that contains all the information necessary to discern the object's outline.

In Data Reduction stage the frame differencing is made by using a special Matlab function. The small objects are removed using the function *bwareaopen* this process is made to eliminate any unnecessary objects. Each object in the image is marked using the *regionprops* Matlab function and the objects are properties are measured using the same function.

In the feature analysis, the features of the objects which are extracted by data reduction process are examined and evaluated for their use in the application.

This stage includes finding the car object in the frames and tracking it through the different frames and calculating the speed of the vehicles.

The recording of the video operation is not included in the model because, it is not part of the computer vision process and it is considered as part of the hardware.

5 Proposed System

The steps of the process are illustrated here with some discussions of each step. The user must make sure that the image processing tool box is installed by typing the command (*ver*), which will display information about the version of Matlab that we are using, and the toolboxes that are installed in the program.

5-1 Read the Video

After recording the video in the Matlab workspace, the video must be read using (*aviread*) function which read the video from Matlab workspace. The past function doesn't display the video in any form; its only work is to prepare the video for the next steps.

5-2 Convert the Video Frame to an Image

The function *frame2im* converts the single movie frame into an indexed image and an associated color map. A simple loop is used to repeat this operation for the entire movie, the dimension of the loop is decided by the function *length* which returns the total number of frames in the movie. This will prepare the images for further processing.

5-3 Compute the Images Absolute difference

The function *imabsdiff* subtracts each element in current array (current image) from the corresponding element in the previous array (previous image) and returns the absolute difference in the corresponding element of the output array. The output image has the same class and size as current and previous images. As shown in fig (2). The expected output of this process is the vehicle object only.

Figure (2): Compute the Images Absolute difference.

5-4 Convert the image to Black and white

The image is converted to black and white by using the function *im2bw,* which converts an image to binary image, based on threshold, the threshold is computed based on a function named *graythresh* automatically which computes a global threshold. A threshold that can be used to convert an intensity image to a binary image combined with *im2bw*. Each pixel takes a value that is a normalized intensity value that lies in the range [0, 1].The graythresh function uses Otsu's method, which chooses the threshold to minimize the intraclass variance of the black andWhite pixels.

The results are shown in fig (3).

Figure (3): Convert the image to Black and white

5-5 Label the objects in the Image and Find the Center of the vehicles to keep track of them:

All the components in the image must be marked, By doing this the number of labeled objects is found in (*numobjectsbase*) parameter, this step is related with another step to measure the all the properties of the image components such as the area of the components and the center of each components.

Since the car is the largest object in the image if we call the (*centroid*) property of the (*regionprops*) function, we can obtain the center of the vehicle object.

5-6 Convex Hull Extraction

After the converting to black and white above, moving edges are filled and appear as solid moving blobs. To characterize the blobs, we use a convex hull to approximate the contour of the vehicles. In many cases, a convex hull is a good approximation of the projection of a car. We call the (*ConvexHull*) property of the (*regionprops*) function.

5-7 Bounding Box Extraction

To obtain scaling information directly from the image rather than using explicit

camera calibration, we exploit the known geometric relationships in the images. We do this by constructing a bounding box to enclose the convex hull. This bounding box is used to isolate the area of interest in the image and is similar to window (or key region) processing. This can be easily made by calling the (*Bounding Box*) property of the (*regionprops*) function.

5-8 Geometric analysis

The direction of motion of each vehicle is fixed, the best fit line is computed through the centroids of the convex hulls found in a series of images .

Ground truth distance is estimated by using scale information along the direction of motion, and these distances, with the frame rate of the video sequence, are used to estimate speed. The function *bwdist* computes the distance between the centers of the convex hull.

To get the scale information from the images directly, the Matlab function *bwdist* computes the Euclidean distance transform of the binary image directly. The Euclidean distance between (x1,y1) and (x2,y2) is :

$$\sqrt{(x_1-x_2)^2+(y_1-y_2)^2} \quad \text{..... (1)}$$

The first point (x1,y1) and the second point (x2,y2) are the centers of the vehicle object in first and last points of a line that passes through the centers of the vehicle object in the different frames, the centers is found using the *centroid* property of the (*regionprops*) function. When the vehicle travels a predefined fixed distance the number of frames that is necessary to travel that distance in the image is calculated and it is compared to a ground truth which is the number of frames that is necessary to vehicle to travel the same distance in order to achieve certain speed.

For example if the number of frames that is needed to travel the fixed distance in a certain speed is 80 frames, we can estimate the speed from the following table (1) which presents a pre-calculated number of frames at a different speeds:

	Speed(Km/h)	No. of frames
1	25	110
2	30	98
3	35	87
4	40	77
5	45	65
6	50	54

Table (1): A pre-calculated speed versus number of frames

(Note: This table is a part of a larger table that gives the speed versus the number of frames for each speed from (25-130) Km/h)

Each of the parameter in table(1) is entered in a separate 1D matrix, named Speed and Frames and each new speed can be estimated by calculating the

number of frames can be used to get the speed by using interpolation, the Matlab function (*interp1)* can be used to achieve this process. And for a car that travel the fixed distance in 80 frames the estimated distance using interpolation is expected to be 38.5 Km/h.

If the work is made manually by programming and not by the Matlab builtin functions, the triangular relationship within the bounding box must be exploited. The pixel length, L_pixel, of a vehicle is estimated along the best fit line (L) indicating the direction of travel. It is estimated to be the length of the cord along the best fit line that intersects the bounding box.

***L_pixel=box_width/sin* ……. (2)**

Where ¸ the direction of motion and in this research is equal to 90 degrees, so that the pixel length is equal to bounding box width¸

To map the pixel length to ground truth vehicle lengths, The empirical vehicle length distribution is used. This allows the user to obtain the ratio of the physical length, L_physical, and the pixel length, L_pixel, which is the scale factor s:

S= L_physical/ L_pixel(ft/pixel)…..(3)

The travel distance between frames and the vehicle speed is estimated using the scale factor just obtained from the above geometric analysis. Assuming that distance traveled by a vehicle is defined by the displacement of its centroid.

To obtain the physical distance of moving centers, the scale factors at each pixel along the travel path having angle of ¸ is estimated. To this end, the total number of pixels along the travel path needs to be estimated, which can be obtained by using:

***#of pixels along moving angle =#of vertical pixels/*sin¸˜ .**

Where the number of vertical pixels is simply the vertical pixel length between the first and the last centroids.

For an image sequence with k frames, where s1 is the scale factor at the centroid of the convex hull of the vehicle of interest in the first frame, sk is the scale factor at the centroid of the convex hull of the vehicle in the k-th frame, and the number of pixels along the driving Path between these two centroids is n; the scale change per pixel, s, ca be computed as:

s=s k - s1 /n-1 (ft / pixel2) ……..(5)

The total distance D traversed between the images is then obtained by summing up the scale factor series as:

D= n*((s1+sk)/2)……..(6)

The speed is then estimated as the ratio of the interframe travel distance and the known frame rate.

6 Results

Testing of the system will first begin with loading artificial or acquired images to the software with a certain result expected. Testing will be as simple as a car traveling in front of the laptop camera and its motion viewed on the screen as shown in figure (1).

During the testing, 40 different videos are fed to the system (27.5 hours of video containing in total of 110,918 frames) have been captured, the frame rate was 29 frame per second, each frame was 160*120 pixel in size. Each video contains many cars, and each car speed is measured by the program individually.

The drivers of the vehicles were asked to drive at a certain velocity and the number of frames at each velocity is taken and it is used to make a lockup table for vehicle speeds that ranges from (25-130) Km/h that table looks like table(1) but with much more values, this table is used as reference for each new value measured by the program for the new recorded video. Some examples of the program execution is given in table(2).

The automatic approach gave a very satisfactory estimated accuracy in vehicle speed of about ±3 km/h error rate.

	M	¤		M	¤		M	¤
1	23	+1	8	54	-2	15	91	+1
2	24	+1	9	59	-1	16	94	+3
3	29	+1	10	64	-3	17	101	+2
4	33	-2	11	71	-2	18	106	-3
5	41	+1	12	76	+1	19	111	+1
6	43	-3	13	83	+2	20	117	-2
7	49	-1	14	84	-2	21	119	+2

Table (2): Speed Estimation (Km/h), M: is the manually measured speed. ¤: is the error in the automatic measurements.

The results of measuring the speed of a vehicle that appears in the video can be found the Matlab command window.

7 Discussions

An efficient image processing techniques are applied to estimate travel speed from image sequences of moving vehicles. Simple geometric relations are obtained directly from the image itself and are used without explicit camera calibration. Furthermore, the techniques presented are validated against ground truth by field trials.

It was found that the proposed method is applicable and can provide the means to estimate car speed that can be used by traffic police in suburban roads, the approach used is very accurate when compared to manual measurements (table1), the proposed method provide a cost effective method with a good response time when compared to the specialized machine for estimating vehicles speed (like radar). The proposed method cost is limited to a laptop with a camera and the program, while the radar, for example, can cost up to five thousand dollars for the single device.

8 Graphical User Interface (GUI) for the Program

The Graphical User Interface is shown in figure (4) below:-

Figure (4): Graphical User Interface

Each one of the buttons when pressed will calculate the speed of the vehicle associated with that sample, for instance, if we press button named 50 Km (which means that the vehicle in that sample travel at speed 50 Km (pre-calculated).

After we select one of the samples the (10 Km-130 Km). Images containing the image absolute difference and the same image after it is converted to black and white to illustrate the work of the program, the estimated vehicle speed appear on the Matlab work space.

9 Conclusions

1-The proposed system does not need a professional operator to work, it has a simple GUI.

2-The system cannot tolerate errors in the capturing the video, errors like rotating the camera during the video recording or changing the position of the camera cannot be corrected by the program.

3-The program assumes that there is one vehicle in the street, and it does not

change lanes, so the software is not designed to be used in crowded streets or busy highways, the system best usage is in suburban roads.

4-The system is tested in day-light; no attempt is made to test the system in the nights.

5- The system is tested in a sunny day and no attempt is made to test the system in the different weather conditions (rain or dust).

10 Future Work

1- Increase the number of lanes that the system must cover.

2- Change the angle of view for the video to top view.

3-Change the system so that it can operate in nights as well.

4-Increase the number of cars that can be processed at the same time.

11 References

1- Lazaros Grammatikopoulos, George Karras, Elli Petsa, (2005). "Automatic estimation of vehicle speed from uncalibrated video sequences". International symposium on modern technologies, education and professional practice ingeeodesy and related fields, PP (1-5).

2- D.J. Dailey and L. Li. (1999). "An algorithm to estimate vehicle speed using uncalibrated camera", IEEE, VOL, 4, NO.8, PP (3-12).

3- Jared Friedman. (2005). "Finding Mean Traffic Speed in Low Frame-rate Video" IEEE Trans. Intelligent Transportation Systems Vol, (1), No. 2, pp. (98-107).

4- Masakatsu Higashikubo(1) Toshio Hinenoya(2) Kouhei Takeuchi(3).2001. "Traffic Queue Length Measurement Using an Image Processing Sensor". In Journal of Further and Higher Education, VOL,25,NO.(2).CarfaxPublishing, PP(10).

5- Rafael M.crouch."The Matlab image processing toolbox guide",USA,Adison-Wisley, 2004,PP(116-150).

6- Rafael C. Gonzalez and Richard E.woods,"Digital Image Processing", Adison-Wesley, USA 2002, PP (20-80).

7-John K. Wally. "The Matlab image processing toolbox guide", 2004.Prentice hall USA, PP (50-70).`

معالجة الصور لتحديد سرعة المركبات

م.م. عمار عوني عباس

جامعة بغداد

المستخلص:

يستخدم هذا البحث معالجة الصور لمحاولة تحديد سرعة المركبات التي تتحرك اما الكاميرا التي تلتقط صوراً فديوية متسلسلة (اطارات) للشارع والسيارات التي تسير فية وفي الزمن الحقيقي. نقوم اولا بتحويل الصورة الى صورة احادية الالوان(اسود وابيض) ثم نقوم باستخلاص الحافات ونصنف الصور الناتجة لنحدد الاجسام الموجودة في الصورة وبالتالي نحدد المركبات ونقوم بملاحقة هذه المركبات في الاطارات المختلفة ونستطيع تحديد سرعة السيارة بحساب موقع جسم السيارة في الاطارات المختلفة.يستخدم هذا البحث طريقة جديدة لتحديد السرعة فبدلا من استخدام شواخص في الطريق لتحديد السرعة فاننا نستخدم دوال برنامج الماتلاب والعلاقات بين الاجسام المختلفة المستخلصة من الصورة نفسها. هذه الطريقة لاتحتاج الى ضبط للكاميرا كما في الطرق المستخدمة حاليا كما انها لاتحتاج الا الى كاميرا بسيطة مرتبطة بحاسبة ويمكن استخدامها بواسطة شرطة المرور لمراقبة السرعة في الطرق الخارجية والداخلية ,كما يمكن ان ندمجها مع نظام لقراءة وتحديد ارقام السيارات لتكون نظاما متكاملاً لاصدار مخالفات السرعة الياً للسواق الذين يتجاوزون السرع المحددة. تم اختبار النظام في طرق مختلفة واكدت النتائج مصداقية الطرق المستخدمة وقدرة النظام على العمل في مختلف الظروف .