Speed Detection Camera System using Image Processing Techniques on Video Streams

Osman Ibrahim, Hazem ElGendy, and Ahmed M. ElShafee, *Member, IEEE*

***Abstract*—This paper, presents a new Speed Detection Camera System (SDCS) that is applicable as a radar alternative. SDCS uses several image processing techniques on video stream in online -captured from single camera- or offline mode, which makes SDCS capable of calculating the speed of moving objects avoiding the traditional radars' problems. SDCS offers an en-expensive alternative to traditional radars with the same accuracy or even better. SDCS processes can be divided into four successive phases; first phase is Objects detection phase. Which uses a hybrid algorithm based on combining an adaptive background subtraction technique with a three-frame differencing algorithm which ratifies the major drawback of using only adaptive background subtraction? The second phase is Objects tracking, which consists of three successive operations, Object segmentation, Object labelling, and Object canter extraction. Objects tracking operation takes into consideration the different possible scenarios of the moving object like; Simple tracking, object has left the scene, object has entered the scene, object cross by another object, and object leaves and another one enters the scene. Third phase is speed calculation phase, which is calculated from the number of frames consumed by the object to pass-by the scene. The final phase is Capturing Object's Picture phase, which captures the image of objects that violate the speed limits. SDCS is implemented and tested in many experiments; it proved to have achieved a satisfactory performance.**

***Index Terms*—Image Processing, speed detection camera, radar, object recognition, object tracing.**

I. Introduction

The need to use radar systems is growing in importance. This is not only for military applications but also for civilian applications. The latter includes (but not limited to) monitoring speeds of vehicles on high ways, sport competitions, aeroplanes, etc [1-12].

The spread of use of radar systems is affected negatively with the high cost of radar systems and also with the increasing requirements on the accuracy of the outputs. This motivated the research on alternative technologies that offer both higher accuracy and be more cost effective.

The field of image processing has grown considerably during the past decade. This has been driven by 1) the increased utilization of imagery in myriad applications, coupled with 2) improvements in the size, speed and cost effectiveness of digital computers and related signal processing technologies. Image processing has found a significant role in scientific, industrial, space and government applications. Many systems nowadays can be replaced by image processing alternate systems that perform better than the former systems. An SDCS system is one of these systems that can replace traditional radars. An SDCS system is applicable as an alternative to current radar systems. This is better cost effective system than current ones. It also has accurate outputs as traditional radars or even better. SDCS system can be integrated with Automatic Number Plate Recognition (ANPR) system to form a complete radar system. ANPR system is a mass surveillance method that uses optical character recognition on images to read the license plates on vehicles.

In the literal [1], authors introduce the primary steps towards developing the Speed Detection Radar. Here authors introduce a new approach in object detection technique, which is “adaptive background subtraction” as it proofs that it’s not sensitive to sudden illumination changes. Another feature is introduced here regarding object tracking by developing “object tracking categories”. Fully functional software is developed and captured windows are added in section IV.

The rest of this paper is organized as follows. Section II presents the issues and system analysis of SDCS. Section III presents the novel design and theory of the system along with its implementation. Developed software snapshots are presented in Section IV. Section V concludes this paper. The system algorithms flowchart is presented in the Appendix.

II. ssues and System Analysis

This section discusses the problem definition, studying of traditional radar systems, then requirements and analysis of the proposed SDCS.

A. Problem Definition

Nowadays radars are extremely expensive. Nevertheless, their accuracy falls short of several potential applications. Consequently, they need to be replaced by automated system in order to have better accurate outputs, less expensive systems, and exclude human factor from the system.

Recently there are two types of radars commonly used in Egypt:

- High way radars: these radars are extremely expensive (about 200,000 – 300,000 LE). They calculate the speed of moving vehicles by means of sensors and capturing still image for vehicles violating limited speed.
- Inner town radars: these radars are less expensive (about 70,000 LE). they calculate the speed of moving vehicles

Manuscript received May 4, 2011; revised August 5, 2011.

Osman Ibrahim is with Informatique Corp, Cairo, Egypt, (e-mail: DrOsman@Yahoo.com).

Hazem ElGendy is with the Computer Science Dept., Ahram Canadian University, 6th October city, Giza, Egypt (e-mail: H_elgendy@masrawy.com).

Ahmed M. ElShafee is with the Network Dept., Ahram Canadian University, 6th October city, Giza, Egypt (e-mail: aelshafee@ieee.org).

by means of sensors only and it needs an operator to capture the images for the vehicles violating speed limit.

- Other technologies are being used to avoid detection by traditional radars; includes:
- Laser detectors and radar detectors, which detect when the vehicle's speed is being monitored and warn the driver (these may be illegal in some areas).
- Laser jammers and radar jammers, which "jam" the laser and radar by returning a scrambled signal which the speeding camera cannot process (these may be illegal in some areas).

Thus, the need for another system to replace the traditional radars due to the disadvantages has become a necessity.

B. Systems Survey

B.1 Gatso

There are over 4,000 fixed Gatso speed cameras currently in use by police forces and local authorities across the UK [11]. The cost of installing a Gatso speed camera is approximately £20,000, but cost can go as high as £40,000 if located in a rural location, as the system requires a 240v power supply.

The fixed Gatso camera has the ability to take up to 400 pictures. Gatso speed cameras can also distinguishes between cars/vans and HGVs separately.

B.2 Truvelo [11]

Truvelo system uses in the road, used to gain the vehicles speed and infrared flash light instead of visible flash light..

B.3 SPECS[11], [13]

SPECS average speed camera systems utilize state of the art video system with Automatic Number Plate Reading (ANPR) digital technology. Consisting of a minimum of two cameras each fitted with infra red illuminators fitted on gantries above the road.

B.4 Peek [11]

Peek relies on radar technology, similar to a Gatso. They are also rear-facing due to the 'flash'.

C. Primary Analysis

After intensely analysis our problem domain and reviewing similar systems available nowadays, It was noticed that SPECS system is the nearest one to our system available nowadays. The main difference is that SDCS use only one digital camera and do analysis on videos in online or offline mode. The basic features that our system should include the ability to calculate the speed of moving objects avoiding the traditional radars' problems.

D. Requirements and Analysis

Requirements may be either functional or non-functional requirements.

Functional requirements specify the functionality that a system or component provide.

- Supporting online and offline modes.
- Categorization of inputted video in offline mode into predefined extension (avi).
- Categorization of the camera to provide an online video stream in online mode.
- Specify the distance captured.
- Specify the limited speed for objects.
- Specify the folder to save the images in.
- Detecting moving objects.
- Tracking moving objects.
- Shadow removal.
- Calculating speed of objects entered the scene and exit appropriately.
- Marking the vehicle which violates the limited speed.

The following are some of the system's non functional requirements:

- Interactive User Interface (GUI)
- Troubleshooting for system inconsistencies
- Portability
- Performance Requirements(speed, response time)
- Reliability of system with its functionality
- Manageability and Ease of Use of the system main features
- Scalability
- Extensibility
- Robustness

III. System Design, Research and Implementation

This section briefly discusses SDCS novel theory, system detailed design and project implementation.

SDCS can be divided into four successive phases; these are

- Objects detection
- Objects tracking
- Speed calculation
- Capturing Object's Picture

Detection of moving objects in video streams is known to be a significant, and difficult, research problem. Aside from the intrinsic usefulness of being able to segment video streams into moving and background components, detecting moving blobs provides a focus of attention for recognition, classification, and activity analysis, making these later processes more efficient since only "moving" pixels need be considered.

The aim of object tracking is to establish a correspondence among objects or object parts in consecutive frames. It also aims to extracting temporal information about objects such as trajectory, posture, speed and direction. Tracking detected objects frame by frame, in video is a significant and difficult task. It is a crucial part of smart surveillance systems. This is because without object tracking, the system could not extract cohesive temporal information about objects. In such cases, higher level behaviour analysis steps would not be possible. On the other hand, inaccurate foreground object segmentation due to shadows, reflectance and occlusions makes tracking a difficult research problem.

Since the output of object detection phase is considerably reliable and it handles sudden illumination changes and shadows. Therefore the foreground image is ready for segmentation, labelling and tracking. Also, the objects' speeds can be calculated by detecting the first frame which the object has entered the scene at (Fr_0) and keeping track of the object till it leaves the scene at frame (Fr_n). This is while neglecting the other unimportant objects such as people crossing the road.

A. Object Detection

A Hybrid algorithm for detecting moving objects [4] is used. It is based on combining an adaptive background subtraction technique with a three-frame differencing algorithm. This combination ratifies the major drawback of using only adaptive background subtraction. That is it makes no allowances for stationary objects in the scene that start to move. Although these are usually detected, they leave behind "holes" where the newly exposed background imagery differs from the known background model as shown in Figure 1.a

While the background model eventually adapts to these "holes", they generate false alarms for a short period of time. Frame differencing is not subject to this phenomenon; however, it is generally not an effective method for extracting the entire shape of a moving object as shown in figure 1.b.

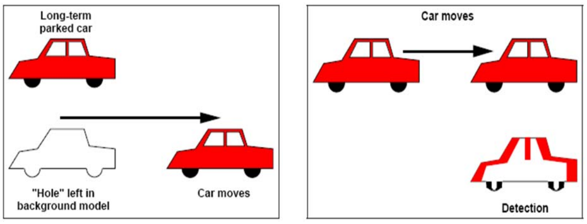


Fig. 1.a. background subtraction leaves holes when stationary objects move. Fig. 1.b. frame differencing doesn't detect entire object

To overcome these problems, we preferred to use the combination of the two methods.

Object detection process, contains three different successive steps, these are

(1)Constructing the motion matrix

The main idea of this section is to construct a matrix corresponds to the current frame. This is being processed to decide what pixels are in motion and what are stationary. The moving pixels will have high probability to represent a foreground pixel while stationary pixels will have high probability to represent a background pixel.

A three-frame differencing operation is performed to determine regions of legitimate motion as shown at figure 2 (we call that part "constructing the motion matrix"). It is followed by adaptive background subtraction to extract the entire moving region. Consider a video stream from a stationary (or stabilized) camera. Let In*(x,y)* represent the intensity value at pixel position *(x,y)*, at time t=n. The three-frame differencing rule suggests that a pixel is legitimately moving if its intensity has changed significantly between both the current image (In) and the last frame (In$_{-1}$), and the current image (In) and the next-to-last frame (In$_{-2}$). That is, a pixel *(x,y)* is moving if

$$\begin{aligned}&(|\ln(x,y)-\ln_{-1}(x,y)|>T_n(x,y))\ and\\&(|\ln(x,y)-\ln_{-2}(x,y)|>T_n(x,y))\end{aligned} \tag{1}$$

where $T_n(x,y)$ is a Threshold describing a statistically significant intensity change at pixel position (*x,y*) (described below). The main problem with frame differencing is that at two frames differencing pixels interior to an object with uniform intensity aren't included in the set of "moving" pixels [2]. Also, at three frame differencing is that its being high sensitive to noise. Nevertheless, if we use a low frame rate the difference between three successive frames will increase and if we use a high frame rate three frame differencing will act as two frame differencing with adding slightly more details.

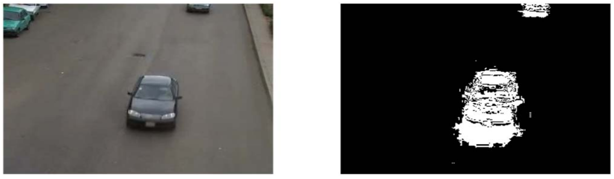


Fig. 2. Motion matrix sample, a; sample scene with two moving objects, b; the constructed motion matrix represents the pixels which are in motion in the current frame ln.

(2)The masked subtraction

Let $B_n(x,y)$ be the corresponding background intensity value for pixel position (*x,y*) estimated over time from video images I0 through In$_{-1}$.

The foreground image *F*(*x,y*) can be formed by the formula:

$$\begin{aligned}F(x,y)=1\ \ &if\ |\ln(x,y)-B_n(x,y)|>T_n(x,y)\\0\ \ &otherwise\end{aligned} \tag{2}$$

This formula is being used by most of the background subtraction techniques. Our new formula is based on this formula and motion matrix which we discussed previously (sometimes referred to as the mask).

This motion matrix can be used as a mask while subtracting the current frame In(*x, y*), the background B_n(*x, y*).

The idea is that, if the pixel is not within the moving area, i.e. not in motion therefore we don't need to apply background subtraction on it.

In other words, if the pixel is not moving so we are going to neglect it since the probability of stationary pixel to be a part of an object is so low while the probability of another pixel which is in motion to be a part of object is so high.

Ignoring other pixels will cause some loss of information, and will lead to un-connected objects. But, we combine the output of this operation with the result of the two frame temporal differencing (temporal differencing detect about 35-50% of the object). This is proven to give a very impressive result as shown in figure 2.

So the foreground image *F*(*x, y*) can be formed by using our masked subtraction method, and the two frame differencing method.

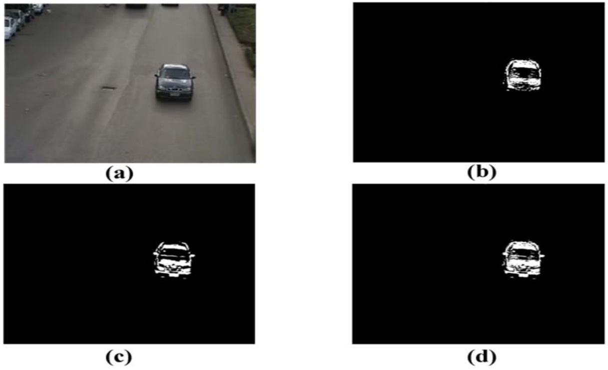


Fig. (3). Masked subtraction sample, a: sample scene with moving object, b: foreground image constructed by masked subtraction, c: foreground image constructed using two frames differencing, d: final foreground image after combining both images

Most of the foreground detection algorithms are susceptible to both shadows and sudden illumination changes which cause inaccurate foreground object segmentation. Since later processing steps like object classification and tracking depend on the correctness of object segmentation, it is very important to cope with shadow and sudden illumination changes in smart surveillance systems.

After examining the properties of shadow one concludes that:

1. Shadow regions are darker.
2. Shadow regions represent the same background surface under a reduced illumination, and share similar textures to the background.

There two approaches for shadow detection and removal, (1) using edge detection, (2) compare the intensities of pixel In(*x,y*), $B_n(x,y)$.

The drawback of the first approach simply is complexity, it is a time consuming approach but give an accurate results and can be used to differentiate between self shadow and cast shadow, but since our application main approach is speed while keeping quality in mind, we decided to use the second approach.

Simply we need a robust and adaptive method to detect whether the current pixel In(*x,y*) is a foreground or shadow. After try and test we have concluded that:

If $0.23 < \text{In}(x,y)/B_n(x,y) < 0.95$ then the pixel is shadow

Otherwise the pixel is a foreground.

The strength of this method is that: it is simple and adaptive to a wide range of shadows. We keep the thresholds at those two ranges. This is in order not to mess-detect a shadow point as if it is a foreground one. Although this can affect the object at some special cases, like the dawn period, it still does not mess-classify shadow. It will just detect a smaller part of the object but with no shadow at all.

As a result we have constructed a foreground image without any shadow. It is adaptive to sudden illumination changes (since we use adaptive background). Besides that, we are using a dynamic threshold which will adapt to the lighting conditions.

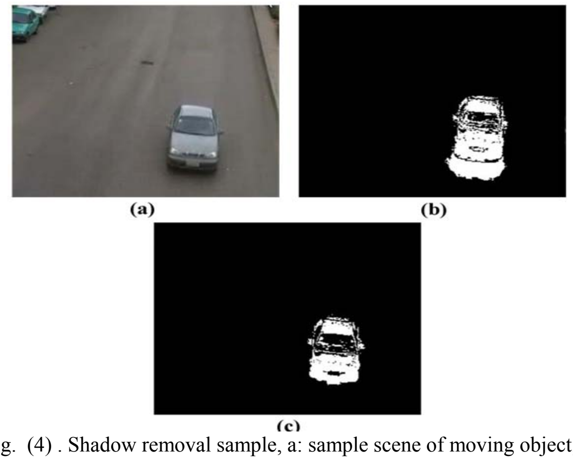


Fig. (4) . Shadow removal sample, a: sample scene of moving object, b: foreground image constructed by masked subtraction before shadow removal, c: the foreground image constructed by masked subtraction after shadow removal.

(3)Generating new background Threshold matrix

Both the background model $B_n(x,y)$ and the difference threshold $T_n(x,y)$ are statistical properties of the pixel intensities observed from the sequence of images $\{I_k(x,y)\}$ for $k<n$.

$B_0(x,y)$ is initially set to the first image, $B_0(x,y) = T_0(x,y)$, and $T_0(x,y)$ is initially set to a pre-determined value which is 15.

B(*x,y*) and *T*(*x,y*) are updated over time by using these equations:

$$\begin{aligned} B_{n+1}(x,y) &= \alpha B(x,y) + (1-\alpha) In(x,y) \quad \textit{moving} \\ &= B_n(x,y) \quad \textit{fixed} \end{aligned} \tag{3}$$

$$\begin{aligned} T(x,y) &= \alpha T_n(x,y) + (1-\alpha)(5\times|In(x,y) - B(x,y)|) \quad \textit{moving} \\ &T_n(x,y) \quad \textit{Fixed} \end{aligned} \tag{4}$$

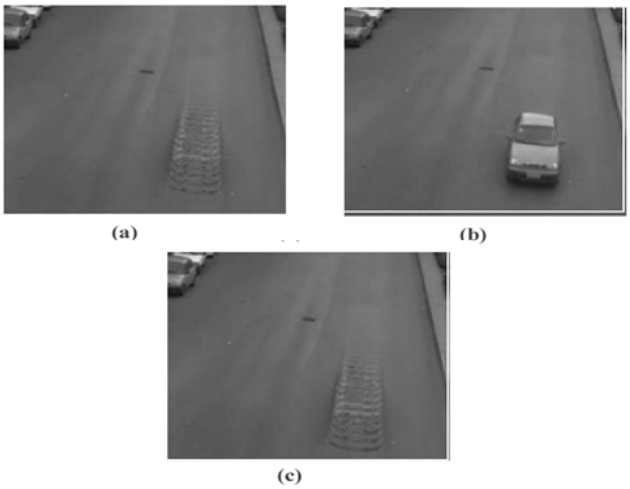


Fig . (5). new background generation, a: initial background, b: current frame, b: new generated background.

B. Objects Tracking

Tracking detected objects frame by frame in video is a significant and difficult task. It is a crucial part of smart surveillance systems since without object tracking, the system could not extract cohesive temporal information about objects and higher level behaviour analysis steps would not be possible. On the other hand, inaccurate foreground object segmentation due to shadows, reflectance and occlusions makes tracking a difficult research problem.

Object tracking driven using three successive phases;

- Object segmentation
- Object labelling
- Object center extraction

(1)Object segmentation

Object segmentation is based mainly upon the connectivity of the objects. In other words, in order to segment a foreground image into a group of objects we must assure that every object is being connected as one part. Otherwise, segmentation will not act in an appropriate way; this will result in excess objects count since the single object is being treated as many several objects.

Since the objects are not connected, we need to detect the area which surrounds the objects. Later, this area can be treated as an excellent representative to the un-connected objects; i.e. this operation aims to map the object into a rectangle representing it.

The method consists of successive iterations; each iteration consists of two main parts, horizontal scanning and vertical scanning.

The horizontal scanning starts from the top most left pixel at the foreground image F (0, 0) Then, it scans the foreground image horizontally. If no foreground pixel (white pixel) found then, it will mark the whole scan line. Else, if any pixel found to be a foreground pixel (i.e. colored in white) then, the scanner will skip this line and go the next scan line, Figure 5.1a show clearly the result after horizontally scanning the foreground image.

On the other hand, the vertical scanning starts also from the top most left pixel at the foreground image F (0, 0). But, it scans the foreground image vertically. If no foreground pixel (white pixel) found then, it will mark the whole scan line. Else, if any pixel found to be a foreground pixel (i.e. coloured in white) then, the scanner will skip this line and go the next scan line.

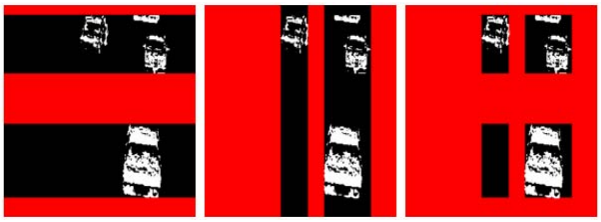

(a) (b) (c)

Fig. (6). Object segmentation 1st iteration

After the first iteration it is clear that we still need another iteration in order to give an accurate result, beside there exist false detected regions which will disappear after an additional iteration as shown below at figure (7)

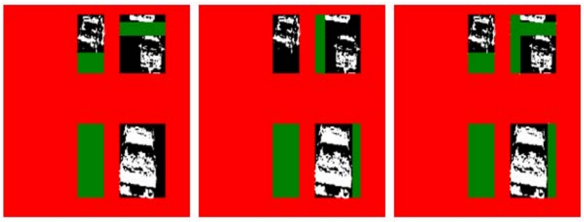

(a) (b) (c)

Fig. (7). Object segmentation 2nd iteration

The second iteration has separated the two objects at the same region shown at figure 7.a and has discarded the false detected regions.

The third iteration is sufficient since no more segmentation can be done, now comes another problem, how to determine the number of sufficient iterations to give an accurate segmentation?

Simply the system is doing segmentation till no more segmentation is available, and thus the method has proved itself to be flexible and adaptive to whatever special cases presented.

Finally this self-developed method has proved itself to be a reliable segmentation technique with very impressive results and with no need to assure object connectivity before dealing with segmentation.

(2)object labelling

In order to keep track of the moving objects, labelling is an essential process. This is because each object must be represented by a unique label while keeping in mind that the object shall preserve its label without any change. This is since the moment it enters the scene (at frame F_0) till it leaves the scene (at frame F_n)

The segmentation process held at the previous stage has guaranteed us a set of well separated regions representing the objects. This is simply since each region represents the object then each region must be given a unique label and preserve it till the object leaves the scene.

Figure 8.a clarifies the output of the segmentation process, it is clear that those regions representing the objects are well separated. At figure 8.b labelling has given each pixel within the same region a corresponding label clarified by a certain colour.

(3) Centre extraction

The object is being ready for the tracking phase. But, for optimization issues, we have discovered that no need to track the whole object pixel by pixel, we just need a descriptive point representing the object.

Simply this point is the object centre; it represents the whole object and can be tracked and mapped easily. In the next section, we discuss in details how can we track the centre and correct its label at some special cases in order to preserve the uniqueness of labelling. For the same object, figure 9 shows the centre of each object which we are going to track at the next stage.

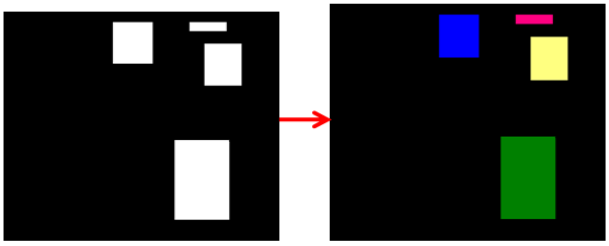

(a) (b)

Fig. (8). Labeling operation

Each Label has been given a unique color for clarification:
Label "1" ➔ Green
Label "2" ➔ Yellow

(4)Object tracking categories

In order to calculate the speed of the moving objects the system need to detect when the object has entered the scene (frame F_0) then it shall track the object and keep it under observation till the object leaves the scene (Frame F_n).

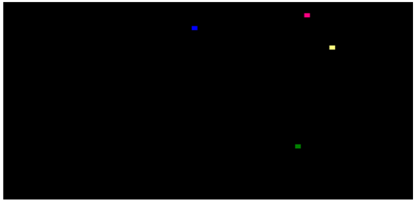


Fig. (9). Center extraction sample

There are three different categories of object tracking

(a) Simple tracking

In this case, the centres of objects, as shown in figure (10.a) representing the previous frame, have moved or shifted for a certain distance from there original position to form the next frame represented at figure (10.b).

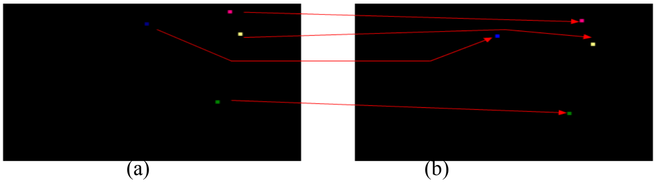

(a) (b)

Fig. (10). Tracking centers within two successive frames *(b) object has left the scene*

When a label L1 (displayed at figure 11.a marked with green), which exist at the frame I_{n-1}, disappears at the next frame I_n.

This indicates that an object has left the scene, therefore the current frame In represents the last frame (F_n) for that green object and therefore the system need to calculate the speed of that left object (this will be discussed later at the speed calculation section).

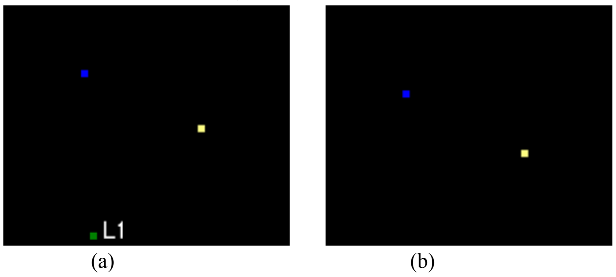


Fig. (11). Label disappears case

(c) object has entered

When a label L (displayed at figure 12.b marked with green) which does not exist at the frame I_{n-1} then appears at the next frame In this indicates that an object has entered the scene, therefore the current frame In represents the first frame (F_0) for that green object and therefore the system creates a record for that new object in order to keep tracking it frame by frame till it leaves the scene.

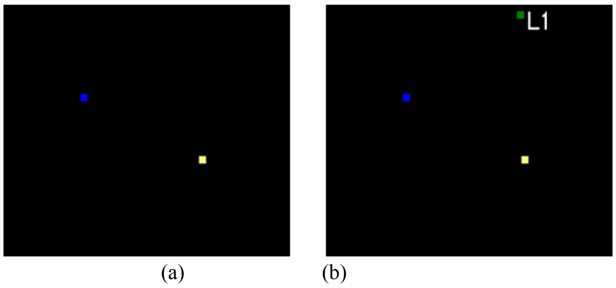


Fig. (12. Label appears case

(d) object cross by another object

Label L1 (displayed at figure 13.a marked with green) is ahead of another label L2 (displayed at figure 13.a marked with Yellow), at the next frame label L2 has become ahead of Label L1.

If this case left un-handled those two labels will swap and will cause an error at the tracking process, but since our tracking use each label history to correct the labelling therefore this swapping would never occur and objects L1 and L2 will preserve there labels as show at figure 13.b

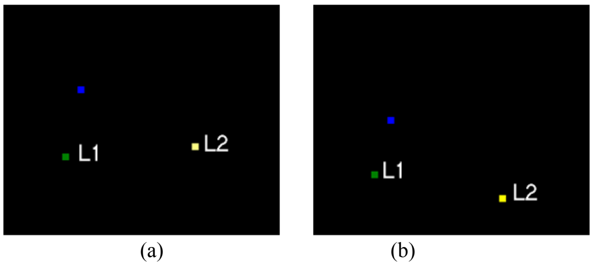


Fig. (13). Label cross by another label case

(e) object leaves and another one enters the scene

If the number of objects at the last frame I_{n-1} equal the number of objects at the current frame In, then there may be no change has occurred and this has been discussed previously at Simple tracking section, but there may exist another case, if there is an object leaves the scene at frame I_{n-1} and another one has entered at the current frame In then that is a special case which need to be handled.

In figure 14, there are three objects in the two frames, but they aren't the same objects, in frame (a) there is an object labelled by one represented by the green colour (L1) (b) a new object entered to the scene, so it will take be given a label four represented by the pink colour (L3) since the green object hasn't released its label yet.

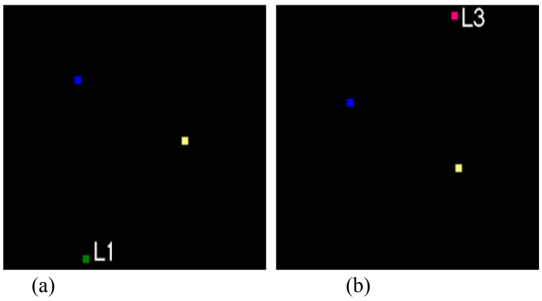


Fig. (14). Label cross by another label case

In order to avoid reserving a certain label for each object even if it leaves the scene, the system assigns labels for every object at a round robin fashion, consider figure 15, at 15.a the green object is still at the frame I_{n-2}, at the next frame In-1 (figure 15.b) the green object has left the scene, at the next frame In (figure 15.c) a new object has entered the scene, and since the label one (represented by the green colour) is not being used at the moment therefore the system will assign the label one to the new object instead of giving it a new label.

C. Speed Calculation

Now after tracking each object in video we can save the frame number that the object entered the scene at (F_{r0}), and the frame number that the object left the scene at (F_{rN}), then speed calculation can be held out by calculating the number of frames consumed by the object to pass-by the scene (enter and leave it) and since we know the duration of each frame (extracted from the video Frame Rate) therefore we can calculate the total time taken by the object to pass-by the whole scene.

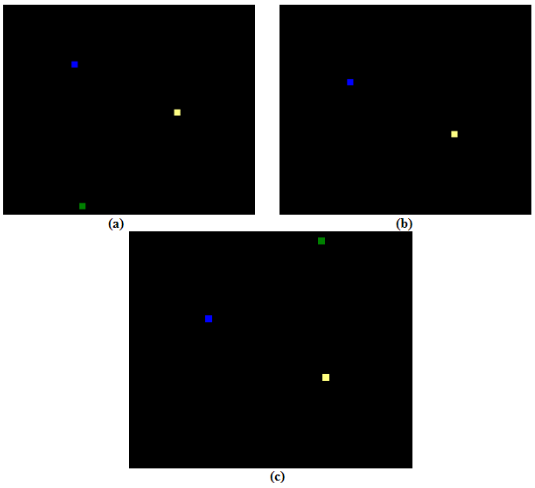


Fig. (15). Object leaves object enter sample with a no-change frame between those events

$$T = N * TF$$

$$N = F_{rN} - F_{r0}$$

T: Total time taken by the object to pass-by the scene.

TF: duration of one frame.

N: Total number of frames.

Then by using the simple equation for speed calculation which states that:

$T= \Delta t = (N * TF)$

Speed $= d / (t_n - t_0) = d / \Delta t$

We can calculate the speed of the moving object easily.

During the tracking phase, the system has stored several information about each object in order to help the system to find the speed:

1- Label ➔ the label that the object takes after labelling operation and correct labels operation.

2- Fr0, FrN.

3- Image ➔Captured Image to the object when it was at the centr of scene.

4- Date, Time of captured image.

After calculating speed, the system check to see if it is violating the pre-defined speed limit determined by the user during the system configuration, if it violates the limited speed then the system stores the object data (picture, speed, Date & Time) and release them from memory (the label also is being released in order to be used by another object at the next frames).

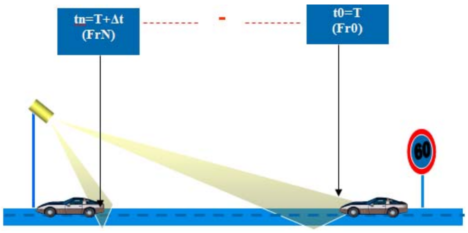


Fig. (16). Speed detection model

D. Capture Object's Picture

The best position to capture a picture with good resolution to the object is that when the object is being at the center of the scene, therefore when the object is located around the center of the scene then the system stores the current frame as the object captured picture.

There exist a wide range of algorithms that work on improving the quality of any image captured from a video stream by capturing three or four consecutive frames (including the target image which needs to be enhanced) from the processed video stream, the new enhanced frame or image is generated after then with better quality than the original one but it needs the video stream to have a high frame rate in order to capture the frames too close from each other. This technique is called video stabilization.

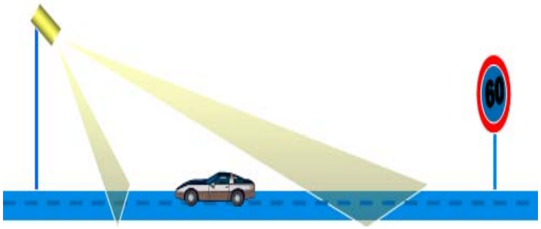


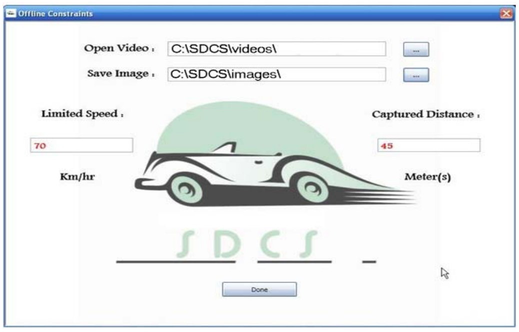
Fig. (17). Object's picture capturing model

After capturing the picture, the system marks the targeted object in the frame in order to differentiate between it and other moving objects within the scene.

Fig. (18). Example for object marking

IV. Developed Software Snapshots

(a) System Configuration



Offline Constraints

Open Video : C:\SDCS\videos\

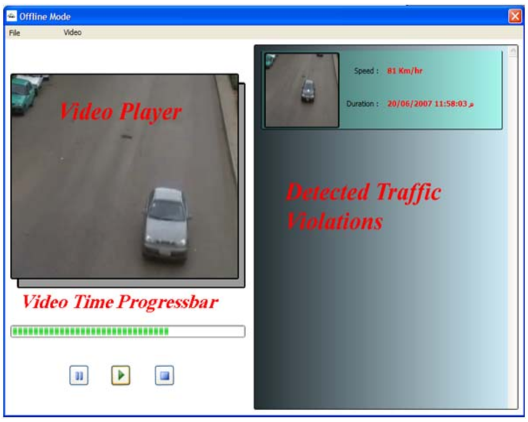
Save Image : C:\SDCS\images\

Limited Speed : 70 Km/hr

Captured Distance : 45 Meter(s)

Fig. (19). SDCC software configuration window

(b) SDCS Main Window



Offline Mode

Speed : 81 Km/hr

Duration : 20/06/2007 11:58:03 م

Fig. (20). SDCC software main window

V. Conclusions

SDCS system provides a software package specifically designed to manage a vehicle's traffic, that provides a number of benefits:

- SDCS is a cheap alternative system to the traditional radar system.
- SDCS is considered as a good application for some difficult image processing algorithms and theories (Object Motion Detection, Shadow Removal, and Object Tracking).
- SDCS doesn't need professional persons to deal with it as it has a simple interface and good design.

References

[1] O. Ibrahim, H. ElGendy, and A. M. ElShafee "Towards Speed Detection Camera System for a RADAR Alternative*," in Proc. 11th International Conf. on ITS Communications*, Sait-Petersburg, Russia, August 2011.

[2] Y. Dedeo˘glu, "*Moving Object Detection, Tracking And Classification For Smart Video Surveillance*," department of computer engineering and the institute of engineering and science of bilkent university, August, 2004.

[3] J. C. Russ, "*Image Processing Handboo*k," the Fifth Edition, 2007.

[4] R. T. Collins, A. J. Lipton, T. Kanade, H. Fujiyoshi, D. Duggins, Y. Tsin, D. Tolliver, N. Enomoto, O. Hasegawa, P. Burt, and L. Wixson, a System for Video Surveillance and Monitoring, "*The Robotics Institut*e," Carnegie Mellon University, Pittsburgh PAThe Sarnoff Corporation, Princeton, NJ.

[5] A. Bevilacqua, Effective Shadow Detection in Traffic Monitoring Applications, "*ARCES.DEIS"* (Department of Electronics, Computer Science and Systems) University of Bologna, Viale Risorgimento, ITALY.

[6] P. L. Rosin, Inst. Remote Sensing App, Joint Research Centre spra (VA), Italy, Image difference Threshold strategies and shadow detection, Tim Ellis, "*Centre for Info. Eng*," City University London, EC1V 0HB, UK.

[7] Y. Matsushita, E. Ofek, X. Tang, and H. Shum, Full- Frame Video Stabilization, "*Microsoft Research Asia Beijing Sigma Center*," No.49, Zhichun Road, and Haidian District.

[8] W. k. Pratt, "*Digital image processing*,*"* fourth Edition, Pixel Soft, Inc. Los Altos, California, Willy.

[9] R. C. Gonzalez University of Tennessee, Richard E. Woods MedData Interactive, "*Digital Image Processing*,*"* Second Edition, Prentice Hall.

[10] R. P. Avery, G. Zhang, Y. Wang, and N. L. Nihan, "*An Investigation into Shadow Removal from Traffic Images*,*"* Department of Civil and Environmental Engineering University of Washington, November 15, 2006.

[11] http://www.ukspeedcameras.co.uk/

[12] http://www.speedcheck.co.uk/

[13] http://www.speedcamerasuk.com/specs.htm

Osman Ibrahim is a recognized figure in software engineering and related technologies in the Egyptian community. For over two decades, he has worked as a practitioner, a researcher, a consultant, a manger, a lecturer, and an author of software engineering and software development.

After receiving a Ph.D. in Computer Science from the Naval Postgraduate School, California, USA, Dr Osman Ibrahim focus was on academia where he taught most of courses (both undergraduate and postgraduate) related to the above focus in reputable Egyptian Public Universities and other private universities. Dr Osman also supervised many graduate thesis and dissertation since then. In the mean time, Dr Osman continued related activities of managing, consulting, and authoring software engineering and related topics.

Dr Osman Ibrahim has headed the Department of Software Engineering in Ahram Canadian University and has published many technical papers in software engineering, data warehousing, and other related topics.

Currently Dr Osman works as Vice Chair and CEO of one of the leading software development companies in Egypt.

Hazem El-Gendy, hold B. Sc. in Electronics and Communications Engineering (Distinction- Honor), his M. Sc. in Computer Eng. and his Ph. D. in Computer Engineering. Dr. El-Gendy worked for Cairo University (Egypt), University of Ottawa (Ontario, Canada), Bell-Northern Research (Canada), Mitel/British Telecom (Canada). Then, he started the consulting firms EPEC Inc. of Canada and GATIS of Egypt. Dr. El-Gendy is currently Assistant Minister of Endowments of Egypt and a University Professor. He has over 170 Internationally published papers (Journals and/or Conferences) and/or contributions to International Standards. Dr. El-Gendy was selected for the 11th Edition of the Marquis' "Who's Who" in Science and Engineering Reference Book, USA.

Ahmed M. ElShafee, Held a Bachelor degree in Electrical Engineering from Faculty of Engineering, Alexandria University, Masters' of science in Electrical Engineering from Faculty of Engineering, Alexandria University, Ph.D. degree in Electrical Engineering from Faculty of Engineering, Alexandria University.

He published many scientific papers, international conferences like Egypt, France, Dubai, Namibia, and India.

He won The Best Young Scientist Award as per the conference council recommendation (National Radio Science Conference 2001), Alexandria, Egypt, for his paper entitled "Rotor Enhanced Block Cipher (REBC)".

He worked in telecommunication engineering field (Operations and Research & Development) for more than 8. Now he works as Assistant Professor, in Faculty of Computer Science and Information Technology (Ahram Canadian University), and as Researcher in Information Technology Research & Consultation Center (ITRCC), Ahram Canadian University.