Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X
IMPACT FACTOR: 6.017

IJCSMC, Vol. 6, Issue. 5, May 2017, pg.137 – 147

ENHANCING PERFORMANCE OF VIDEO BASED VEHICLE DETECTION TECHNOLOGY USING EDGE DETECTION MECHANISM

Jyoti[#1], Rupali Malhotra[#2]

M.Tech student, Department of Computer Science Sat Priya Group Of institutions, Maharshi Dayanand University (MDU), Rohtak

Assistant Professor, Department of Computer Science Sat Priya Group Of institutions, Maharshi Dayanand University (MDU), Rohtak

[1] jyotihooda92@gmail.com, [2] rupaliahuja4@gmail.com

Abstract: Video based vehicle detection technology is an integral part of Intelligent Transportation System (ITS), due to its non-intrusiveness & comprehensive vehicle behavior data collection capabilities. Digital Image processing has been applied to traffic analysis in recent years, with different goals. In recent years, image processing has been applied to field of traffic research with goals that include queue detection, incident detection, vehicle classification, & vehicle counting. This report is explicitly recognizes the speed is an important parameter in traffic analysis system. Relatively few efforts have attempted to measure speed by using video images from un-calibrated cameras. Algorithm for speed extraction first applies a chain of operators to single images to create a set of enhanced images.

Keywords: ITS, MATLAB, EDGE DETECTION, AVI, JPEG, SDCS, GUI

[1] Introduction

The process of vehicle detection and tracking in this work is implemented to conclude corners points and interest points are tracked between video frames using deterministic interest point correspondence method. Based on location & displacement of interest points, vehicle counts & vehicle speeds are determined. The tasks employed to determine above process is explained as follows:

Capture of live video feed: Live video feeds from CCTV cameras monitoring freeways & arterials are captured for video frame processing by a USB frame capture device. The video feeds are captured at various locations at different time of day. Also, CCTV

Cameras are pan-tilt-zoom cameras with varying camera field of view. Height of camera mounted is unknown.

Fig 1 Vehicle Detection & Tracking

Pre-processing of video frames: Using a GUI tool developed as part of our vehicle detection system, user could select region of interest on captured video frame. The detection & tracking algorithms are only performed on this cropped image region to reduce processing time of system. The user is required to specify detection & speed zones using horizontal virtual reference lines. The detection zones are areas where interest points are evaluated, vehicles detected & vehicle counts are incremented. The speed zones are adjacent to detection zones, where interest points are re-evaluated & vehicles are detected. As a rule of thumb, detection zone length should be less than vehicle length as seen in video feed & speed zone length should be just greater than vehicle length as seen in video feed. The user specifies virtual vertical lane reference lines that segment lanes on video frame. These vertical lines are used to determine vehicle counts by path. Also, user specifies direction of vehicle motion or traffic flow, calibration reference line & corresponding distance in physical distance. This reference distance is used to evaluate speed of vehicle. The basic steps for vehicle speed detection are as follows:

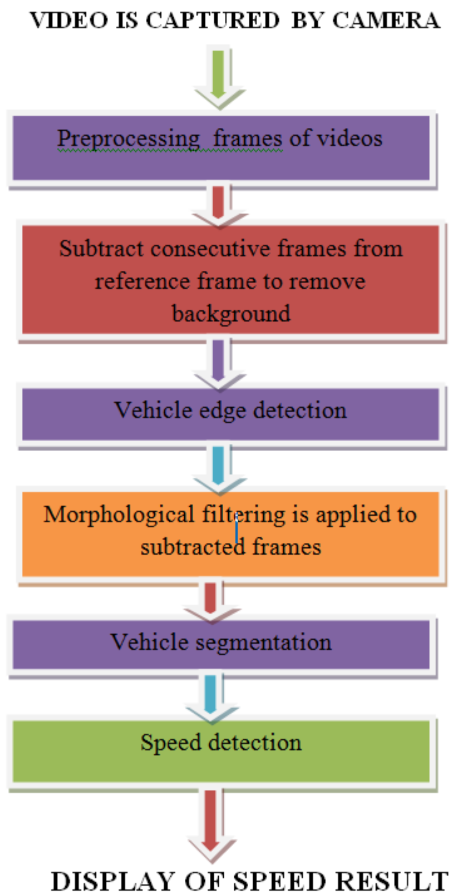

Fig 2 Steps for vehicle speed detection

[2] EDGE DETECTION

Edge detection is name for a set of mathematical methods which aim at identifying points in a digital image at which image brightness changes sharply or, more formally, has discontinuities. points at which image(picture) brightness changes sharply are typically organized into a set of curved line segments termed *edges*. same problem of finding discontinuities in 1D signals is called step detection & problem of finding signal discontinuities over time is known as change detection. Edge detection is a fundamental tool in image(picture) processing, machine vision and computer vision, particularly in areas of feature detection and feature extraction.

Canny edge detection applied to a photograph

Fig 3 Edge detection

CANNY BASED EDGE DETECTION

Canny edge detector have advanced algorithm derived from previous work of Marr and Hildreth. It is an optimal edge detection technique as provide good detection, clear response and good localization. It is widely used in current image processing techniques with further improvements.

Canny Edge Detector

1) Smooth image(picture) with a Gaussian
    - optimizes trade-off between noise filtering and edge localization
2) Compute Gradient magnitude using approximations of partial derivatives
    - 2x2 filters
3) Thin edges by applying non-maxima suppression to gradient magnitude

Detect edges by double thresholding

[3]TOOLS AND TECHNOLOGY

Hardware Requirement

1. CPU (More than 1 Ghz)
2. RAM (2 Gb) Recommended
3. Hard disk (10 Gb free space)

4. Monitor (High resolution)
5. Keyboard
6. Mouse

Software Requirement

1. Windows 7
2. Matlab 2012
3. Java development kit

Matlab is known as Language of Technical Computing. It is considered as a high-level language with interactive environment. Matlab enables us to perform computationally tasks quicker as compare to other programming languages such as C, C++, & Fortran.

Matrix is a rectangular array of numbers in MATLAB environment. Its Meaning is attached to 1x1 matrices. These are scalars. In order to matrices with one row or column there are vectors. The MATLAB has different ways to store numeric & nonnumeric data. It is best to consider everything as a matrix in beginning. Operations in MATLAB have been designed to be natural. Programming languages other than Matlab work with numbers one at a time but MATLAB offers to work with complete matrices quickly & easily.

IMAGE PROCESSING IN MATLAB

Image Types in Toolbox

The Image Processing Toolbox supports four basic types of images:

1. Indexed images
2. Intensity images
3. Binary images
4. RGB images

An indexed image consists of two arrays, an image matrix & a colormap. The colormap is an ordered set of values that represent colors in image. For each image pixel, image matrix contains a value that is an index into colormap. The colormap is an m-by-3 matrix of class double. Each row of colormap matrix specifies red, green, & blue (RGB) values for a single color:

color = [R G B]

R, G, & B are real scalars that range from 0 (black) to 1.0 (full intensity).

The pixels in image are represented by integers, which are pointers (indices) to color values stored in colormap. The relationship between values in image matrix & colormap depends on whether image matrix is of class double or uint8. If image matrix is of class double, value 1 points to first row in colormap, value 2 points to second row, & so on. If image matrix is of class uint8, there is an offset; value 0 points to first row in colormap, value 1 points to second row, & so on. The uint8 convention is also used in graphics file formats, & enables 8-bit indexed images to support up to 256 colors. In image above, image matrix is of class double, so there is no offset. For example, value 5 points to fifth row of colormap.

[4]OBJECTIVE OF RESEARCH

The objective of research is to reduce time consumption during snapshot comparison as well as size of images should be reduced. We captured the video of real traffic and stored in image base. The real distance of the highway that is covered by the view of camera is nearly 15 meter. In video we have single vehicle travelling with the speed of 50 km/hr. But we produced better result as compare to traditional system as we convert frame to images and compare them after finding edges. This reduced the size of image base as well as the time of comparison gets reduced.

[5] RESULT AND DISCUSSION

Comparative analysis of Time consumption during comparison in tradition and proposed comparison system

```
x=[10 20 30 40 50 60 70 80];
y=[3 3 4 4 5 5 6 7];
y1=[1 1 2 2 3 3 3 4];
hold on;
plot(x,y,'r+-');
plot(x,y1,'b+-');
title('Comparative Analysis snapshot comparison delay of Proposed and traditional work');
xlabel('Snapshots');
ylabel('Time Taken Sec');
legend('Traditional', 'Proposed');
```


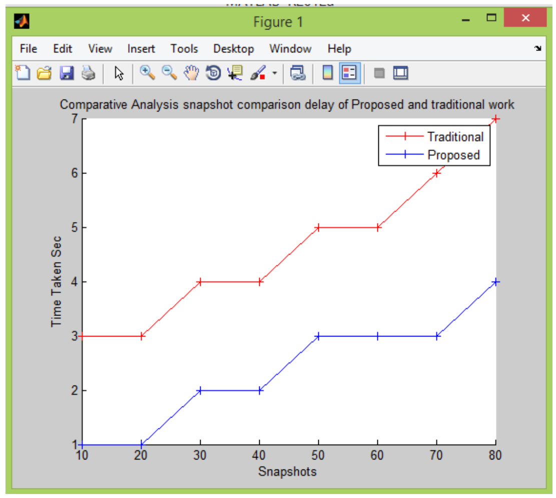

Fig 4. Comparative analysis of Time consumption in tradition and proposed comparison system

Comparative analysis of overall Time consumption in tradition and proposed comparison system

x=[10 20 30 40 50 60 70 80];

y=[5 5 8 8 10 10 11 11];

y1=[2 2 3 3 4 4 5 5];

hold on;

plot(x,y,'r+-');

plot(x,y1,'b+-');

title('Comparative Analysis Processing delay of Proposed and traditional work');

xlabel('Snapshots');

ylabel('Time Taken Sec');

legend('Traditional', 'Proposed');

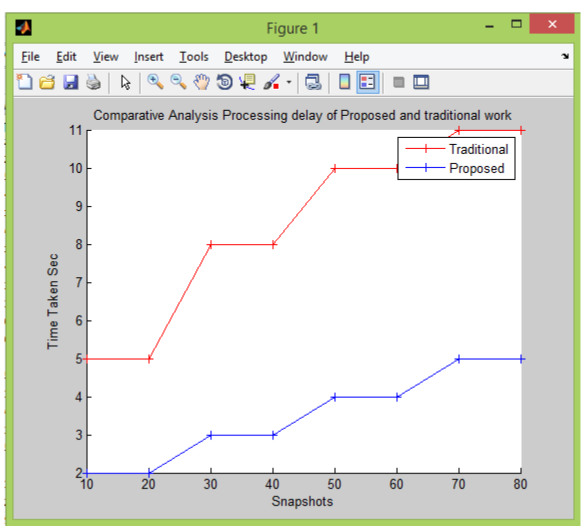

Fig 5. Comparative analysis of overall Time consumption in tradition and proposed comparison system

Comparative analysis of Queuing delay in tradition and proposed comparison system

```
x=[10 20 30 40 50 60 70 80];
y=[6 6 9 9 11 11 13 13];
y1=[3 3 4 4 4 4 5 5];
hold on;
plot(x,y,'r+-');
plot(x,y1,'b+-');
title('Comparative Analysis Queuing delay of Proposed and traditional work');
xlabel('Snapshots');
ylabel('Time Taken Sec');
legend('Traditional', 'Proposed');
```

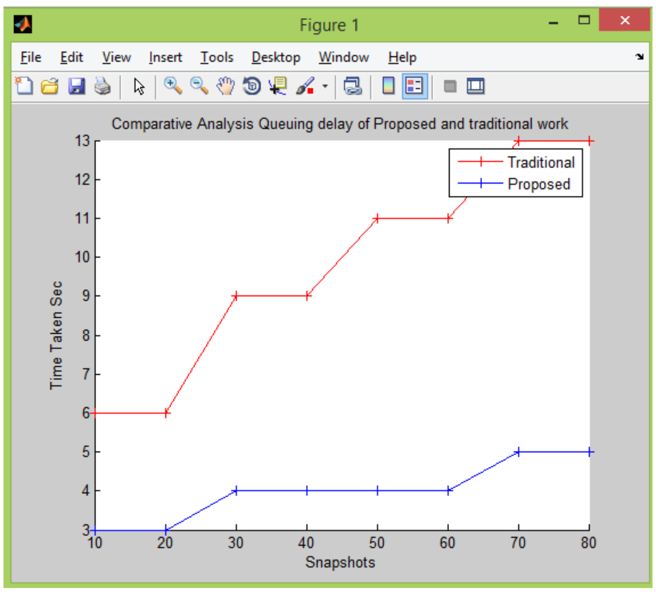

Fig 6. Comparative analysis of Queuing delay in tradition and proposed comparison system

Comparative analysis of File Size in tradition and proposed comparison system

```
x=[10 20 30 40 50 60 70 80];
y=[4020 8090 12100 16201 20300 24200 29002 33100];
y1=[1020 2050 3600 4201 5100 6300 7210 8543];
hold on;
plot(x,y,'r+-');
plot(x,y1,'b+-');
```

```
title('Comparative Analysis File Size of Proposed and traditional work');
xlabel('Snapshot');
ylabel('File Size Kb');
legend('Traditional', 'Proposed');
```

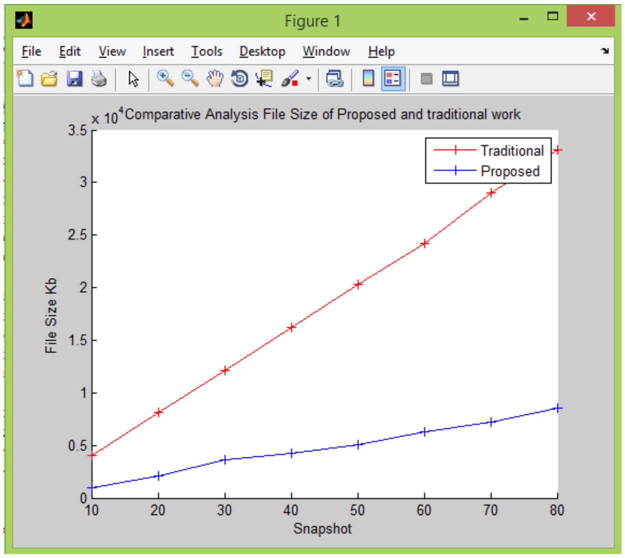

Fig 7. Comparative analysis of File Size in tradition and proposed comparison system

[6] CONCLUSION

Speed is detected for multiple vehicles by processing the video frames. The video taken is taken at high resolution and converted to snapshot. During the mapping process we have also transformed the 3d view of real world coordinates into 2d camera coordinates. We processed video in snapshots and reduced the size of snapshot by applying canny based edge detection mechanism. Then work is done on each frame, here reference frames are converted into grayscale images from RGB that reduce the computation and we are able to detect multiple vehicles simultaneously by drawing the bounding box surrounding to it.

[7] SCOPE OF RESEARCH

There is a research scope in future where we have to accurately detect vehicles in each consecutive frame by drawing bounding box surrounding to it. The video camera-based automatic vehicle speed detection is a very accurate & promising technology for future application in traffic monitoring. It could be a very powerful & cost-effective tool in helping enforcing speed limit law & automatic real-time traffic information reporting.

REFERENCES

[1] - M. Kilger "video based traffic monitoring" in *International Conference on Image Processing & its application*, 7-9 April 1992, Netherlands.

[2] - B. Steux, C. Laurgeau, L. Salesse, & D. Wautier, "Fade: A vehicle detection & tracking system featuring monocular colour vision & radar data fusion," in Proc. IEEE Intell. Veh. Symp., Jun. 2002, vol. 2, pp. 632–639.

[3] - E. Baş, A. M. Tekalp, & F. Sibel Salman, "Automatic Vehicle Counting from Video for Traffic Flow Analysis," Proceedings of 2007 IEEE Intelligent Vehicles Symposium, Istanbul, Turkey, June 13-15, 2007

[4] - kiran kumar, pallavi chandrakant, santosh kumar, kushal, "Vehicle speed detection in video frames using corner detection" *2014 fifth international conference on signal & image processing IEEE*, pp253-258,jeju island©2013

[5] - M. Kilger "video based traffic monitoring" in *International Conference on Image Processing & its application*, 7-9 April 1992, Netherlands.

[6] - Abbas Dehghanj, Ali pourmohammad, "Single camera vehicle speed measurement" 2013 *8th Iranian conference on machine vision & image processing IEEE*, pp190-193, 15-19 Dec. 2007 harbin.

[7] - pumrin, Dailey D.J., "Roadside camera motion detection for automated speed measurement." *intelligent transport sys. Proc., IEEE 5th inter conf. on,* vol no., pp147, 151, 2002

[8] - Budi Sugandi, Hyoungseop Kim, Joo Kooi Tan, & Seiji Ishikawa. Tracking of moving objects by using a low resolution image. In *Innovative Computing, Information & Control, 2007.ICICIC'07. Second International Conference on*, pages 408–408. IEEE, 2007.

[9] - Changick Kim & Jenq-Neng Hwang. Fast & automatic video object segmentation & tracking for content-based applications. *Circuits & Systems for Video Technology, IEEE Transactions on*, 12(2):122–129, 2002.

[10] - Zhan Chaohui, Duan Xiaohui, Xu Shuoyu, Song Zheng, & Luo Min. An improved moving object detection algorithm based on frame difference & edge detection. In *Image andGraphics, 2007. ICIG 2007. Fourth International Conference on*, pages 519–523. IEEE, 2007.

[11] - I. Haritaoglu, D. Harwood, & L.S. Davis. W4: A real time system for detecting & tracking people. In Computer Vision & Pattern Recognition, pages 962–967, 1998.

[12] - C. Stauffer & W. Grimson. Adaptive background mixture models for realtime tracking. In Proc. of IEEE Computer Society Conference on Computer Vision & Pattern Recognition, page 246252, 1999.

[13] - A. J. Lipton, H. Fujiyoshi, & R.S. Patil. Moving target classification & tracking from real-time video. In Proc. of Workshop Applications of Computer Vision, pages 129–136, 1998.

[14] - K Susheel Kumar, Shitala Prasad, Pradeep K. Saroj, R.C. Tripathi **"Multiple cameras using real time object tracking for surveillance & security system "** IEEE 3rd International Conference on Emerging Trends in Engineering & Technology (ICETET), 2010.
[15] - T. Horprasert, D. Harwood, & L.S. Davis. A statistical approach for realtime robust background subtraction & shadow detection. In Proc. of IEEE Frame Rate Workshop, pages 1–19, Kerkyra, Greece, 1999.

