Automatic Vehicle Detection and Road Traffic Congestion Mapping with Image Processing Technique

Pradip Singh Maharjan
Department of Computer and Electronics Engineering
Kantipur Engineering College
Tribhuvan University, Nepal

Ajay Kumar Shrestha
Department of Computer and Electronics Engineering
Kantipur Engineering College
Tribhuvan University, Nepal

ABSTRACT

Road Traffic Congestion Mapping is a system that enables people find information about the traffic congestion in the city. One of the major problems in the city centers like Kathmandu is the traffic jam in the roads. This project provides a feasible solution to the users in finding less congested path on the road to their destination. The system collects traffic congestion data from the roads and makes it available to users via Openstreet map. The surveillance cameras installed at the roads give continuous input to our system which then counts the number of vehicles in the road in a span of time to determine the congestion in the road. The system implements Background subtraction and thresholding for detection of vehicles from the image input received from the cameras. The congestion is plotted in Openstreet map, for example red line for highly congested road, blue line for mildly congested road and green line for free flow of vehicles in the road. Once this information is obtained, one can easily find the alternate path to their destination.

General Terms

Image processing; Vehicle detection

Keywords

Road traffic, congestion, image processing, thresholding, vehicle detection.

1. INTRODUCTION

With rapidly growing urbanization, the need of robust and reliable traffic surveillance system to improve traffic control and management has been rising. Traffic congestion has been increasing worldwide as a result of increased motorization, urbanization, population growth, and changes in population density. Real-time measurement and analysis of road traffic flow parameters such as volume, speed and queue are increasingly required for traffic control and management. Recent development in technologies have shown that traffic detection through real time video, loop detectors, inductive loop sensors and magnetic sensors has been a noble and a promising approach to automatically detect traffic and its parameters[1].

In the main cities of Nepal such as in Kathmandu, road traffic management is done manually by traffic police personnel. Due to frequent traffic jam, people are forced to waste their valuable time getting stuck in congested traffic. Currently no system is there to provide people with reliable information about traffic congestion on the roads.

Kathmandu has few hundreds of surveillance cameras installed around the city by Metropolitan Police Authority Office of Kathmandu. Basically they are used only for surveillance purposes. Our project uses those cameras to detect and count number of vehicles and plot the information on a map so that people can benefit from such information while travelling to their destination.

Information retrieved from surveillance cameras is useful. Manually reviewing large amount of data from video camera is impractical. With the increase in number of cameras the task becomes increasingly difficult. Automatic detecting of vehicles in video surveillance data is very challenging problem [2]. One of the flexible techniques for automatic analysis of road traffic flow parameters is Digital Image Processing which helps in extracting useful information on traffic from video. The system uses Background subtraction technique for detection of vehicles. In this approach, difference between the extracted image frame from surveillance camera and the corresponding background image is performed to detect foreground objects. The result is then processed through Median filter, Binary Erosion and Dilation to remove noise and distinguish moving vehicles from background. The number of vehicles is then counted and interpreted in the Map where users can view and grab the required information more conveniently while visiting the website.

2. RELATED WORKS

Various approaches are being used for automatic vehicle detection from video. Vibha L et al., [2] discussed foreground object detection through difference of consecutive frames at regular interval. Selection of time interval between frames is tricky in this method. Vehicles move at different speed and detection of vehicles could be missed or repeated. If time interval between frames is short, slow moving vehicle appear at same spot and high speed vehicle could be spotted at different positions on image if time interval is long hence producing inaccurate result. Isha et al., [3] presented background subtraction and Sobel Edge detection technique to detect vehicle. In their work, foreground objects were detected by subtracting static background image with input frames. Then the vehicles were classified based on area and circumference using Fuzzy logic. Static background subtraction was affected by weather and illumination changes. D. J. Dailey et al., [4] approached estimation of vehicle speed by using a sequence of images and background subtraction technique in order to detect vehicle. P. Srinivas et al., [5] presented the use of adaptive background technique where new background was generated based on segmentation results extracted from differencing the image with preceding extracted background. Their work presented a technique to eliminate the factors such as weather, change in illumination that affects the results of static background subtraction. A.J. Lipton et al., [6] described vehicle tracking and classification system to distinguish between vehicles and humans.

3. BACKGROUND

It is necessary to assume the stationary background for a given video feed from the camera. Also the algorithm for this does focus on capturing the image frames from the video at fixed time interval. So far detecting the approximate vehicle density is concerned, images captured at interval of 5 seconds is enough to get desired information. The captured frames are to be subtracted from stationary background image to detect foreground objects, i.e. moving vehicles. The difference of the image is then converted into binary image by thresholding and then post processed to filter noises. After this, vehicle is to be detected and counted, and vehicle density can be plotted on map. The architecture of the proposed system is presented via flow diagram as shown in figure 1.

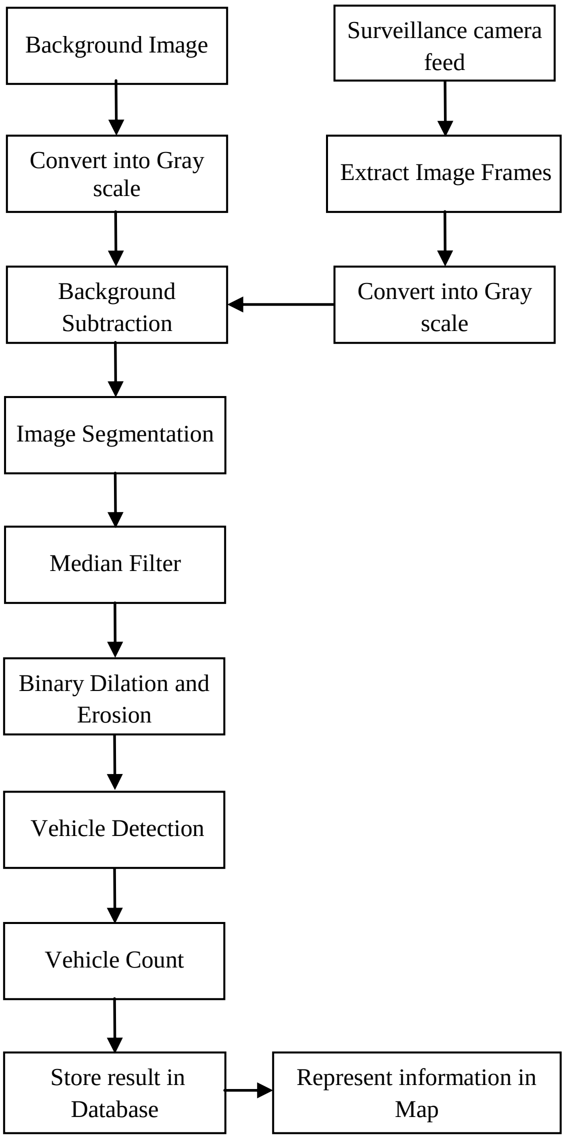

Figure 1: System Architecture

4. PROCEDURE AND RESULTS

First of all, we acquired the video feed from surveillance camera of two locations from Metropolitan Police Authority Office. One of the cameras was positioned at side of the road while the other was directly above the road. The video used for analysis represents calm weather of day time.

In order to detect the vehicles from the video, we applied various image processing procedures. The extracted image frames were subtracted from background image. Then the result of subtraction was passed through thresholding to convert gray scale image to binary image and separate the vehicles from the background. Then median filter was applied followed by binary dilation and binary erosion to remove noises. Then the objects were counted.

4.1 Conversion to Gray scale

After extraction of frames, the image was converted into gray scale so that the processing of the image becomes simpler. Each color pixel of image is described by three intensities of red (R), blue (B) and green (G). Luminosity method was used to convert colored image to gray scaled image. The formula is given below in equation 1.

$$0.21R + 0.71G + 0.007B \qquad (1)$$

The equation is the weighted average to account for human perception. The frames extracted from the video and corresponding conversion to gray scale are shown in figure 3.

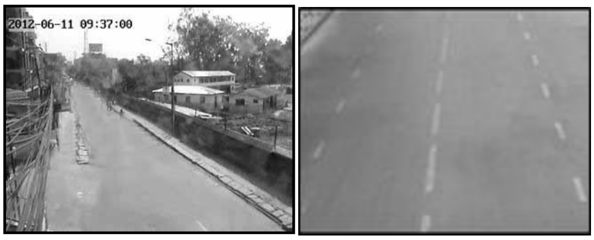

Figure 2: Stationary background image of two different roads

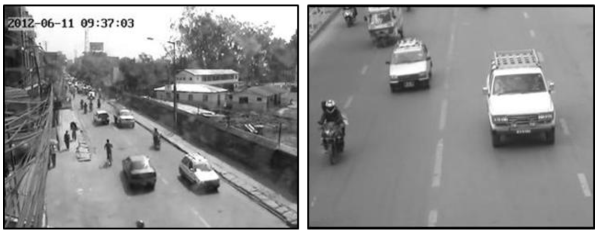

Figure 3: Image extracted from the video and converted to gray scale

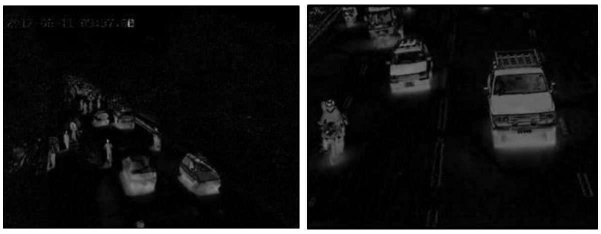

Figure 4: Image output after background subtraction

4.2 Background Subtraction

Identifying objects from a video sequence is a fundamental and critical task in an object detecting application. The static background image without any moving vehicle as shown in figure 2 was subtracted with the image with vehicles as given in figure 3. The output pixel values are given by equation 2.

$$O(x , y) = I(x , y) – B(x , y) \qquad (2)$$

Where O(x, y) is a output image pixel, I(x, y) is input image pixel and B(x,y) is background image pixel.

The result after background subtraction is shown in figure 4. After the background subtraction from the particular image frame, the stationary background was eliminated with black pixels and the pixels of moving objects appear in brighter intensity.

4.3 Image Segmentation

Thresholding method was used to convert the gray scaled image to binary image. As the result shown in figure 4, the background subtraction shows several background pixels that have similar intensity level with object pixels and vice versa. Here multilevel thresholding with two threshold values T_1 and T_2 were used. A point (x,y) belonging to $T_1 < (x,y) \leq T_2$ was classified to object if $f(x,y) > T_2$ and to background if $f(x,y) \leq T_1$[7]. The result after thresholding is shown in figure 5.

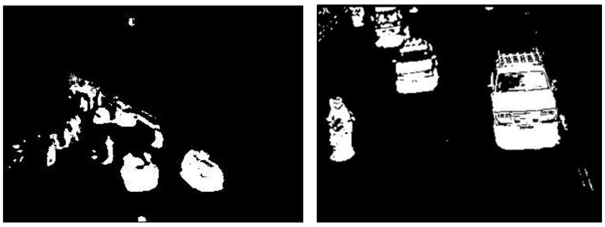

Figure 5: Result after Multilevel thresholding

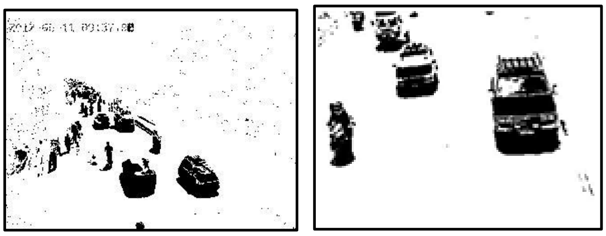

Figure 6: Image output after thresholding and inversion

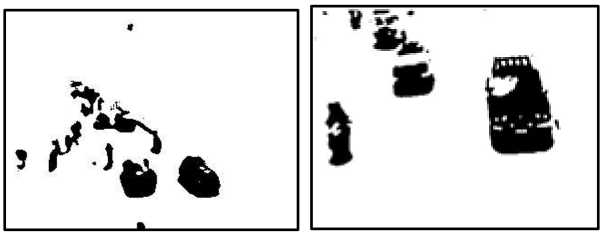

Figure 7: Image output after applying Median filter

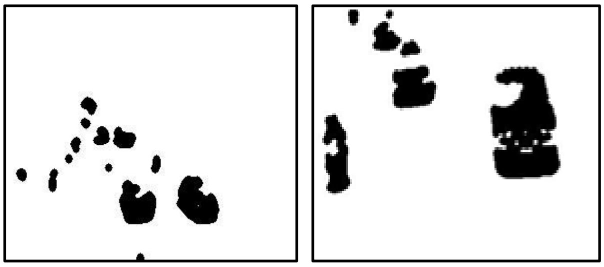

Figure 8: Output after applying Morphological operations

After thresholding, the resulting image was inverted; the black background was converted into white color whereas the white portion or detected objects were represented by black color. The main purpose of this inversion process was to clearly identify the moving objects so that they could be easily post processed thereby assist the object counting. The result is shown in figure 6.

4.4 Median filter

After background elimination and thresholding, there was still some noise. Due to noise and uneven object boundaries, object detection could be erroneous. The algorithm used here was Median filter which is popularly known as order-statistics filter. Median filter as its name implies, replaces the value of a pixel by the median of the values in the neighborhood of that pixel (the original value of the pixel is included in the computation of the median) [7].

After applying median filter to the image containing noise, we got output as in figure 7.

4.5 Morphological Operations

Morphological operations play a key role in applications such as machine vision and automatic object detection. Morphological operations focus on binary images as well as gray scale image [8]. Morphological operations such as Binary Erosion and Dilation are usually performed on binary images, where the pixel values are either 0 or 1. For simplicity, we referred pixels as 0 or 1, 0 for black and 1 for white.

4.5.1 Binary Erosion and Dilation

Binary erosion uses the following for its mask: The result is based on logical AND relationship of the mask on image.

$$\begin{matrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{matrix}$$

This means that every pixel in the neighborhood must be 1 for the output pixel to be 1. Otherwise, the pixel will become 0. Erosion can be used to eliminate unwanted white noise pixels from an otherwise black area. The effect on a binary image is to diminish, or erode, the edges of a white area of pixels.

Binary Dilation process makes the white areas grow. Binary Dilation uses following mask on image with logical NAND:

$$\begin{matrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{matrix}$$

Being the opposite of erosion, dilation will allow a black pixel to remain black only if all of its neighbors are black. This operator is useful for removing isolated black pixels from an image.

The result of applying morphological dilation operation followed by erosion operation can be seen in figure 8. This process was very helpful in separating two vehicles when they appeared to be connected by tiny edges thereby producing better result.

4.6 Vehicle Detection

After morphological operation on the binary image, the moving objects or vehicles were clearly distinguished. Edges of the objects were sharp and smooth hence vehicles could be detected easily and counted. The image was scanned from top to bottom for detecting the presence of Binary Large Objects (BLOB). This was done by counting the number of connected black pixels in the image using following algorithm.

```
For each Pixels(x,y) of image
If Pixel(x,y) is black and not visited before
    Enqueue Pixel(x,y) to Queue q
    Dequeue Pixel(x,y) from q while q is not empty
        Mark Pixel(x,y) as visited
        Count pixels
      Enqueue unvisited neighboring Black Pixels to q
   End while
  If pixelcount>30 and pixelcount<600
        vehiclecount++
Endif
```

Vehicle appearing distinct in image was in the range of 30 pixels to 600 pixels size. Object of size smaller than 30 pixels was due to the noise. The BLOB size of given range was determined as object. The number of vehicles counted on particular road was updated on server database.

Figure 9: Openstreet map with Base layer only

4.7 Plotting information on Map

The information of vehicle density of particular road section was plotted on Openstreet map which is a database of free map with geographic data of Earth under free license. The following things were considered to create a map with OpenLayers requires [9]:

- Included the OpenLayers library files
- Created an HTML element that the map would appear in
- Created a map object from the Map class
- Created a layer object from a Layer class
- Added the layer to the map
- Defined the area of the map

4.7.1 Working in OpenLayers

Layers in OpenLayers are stacked with different features attached to each layer. A layer is basically a way to show multiple levels of information independent of each other [10]. Openstreet map with base or map layer is shown is figure 9. The process to work with layers consisted of two steps:

1. Created the layer object.
2. Added a single layer object to the map. Either a single layer or array of layers could be added to map.

4.7.2 Projections

A projection of 3D coordinates i.e. latitude and longitude of earth on 2D plane of Openstreet map was required. There are some universally agreed classification systems that keep track of projection information. OpenLayers uses European Petroleum Survey Group (EPSG) codes for projection.

We dealt two types of ESPG codes:

1.	EPSG:4326	WGS 1984 Projection
2.	EPSG:900913	Spherical Mercator Projection

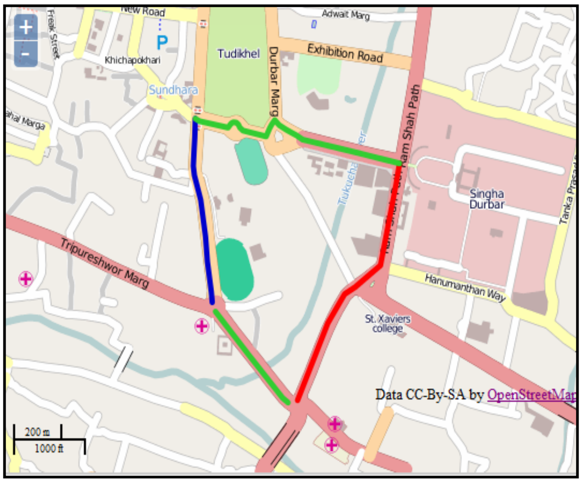

Figure 10: Map after drawing the lines on overlay layer

4.7.3 Rendering to show Congestion Information

Lines of different colors were drawn on map layer to show congestion information. To render a line we performed following steps:

1. Created an array of the points along which the line was drawn
2. Defined styles such as stroke color, stroke width & stroke opacity
3. Created a layer on top of map layer where the rendered line was added

The information about the road traffic was retrieved from database and was displayed on map as in figure 10. Green line indicated free flow of vehicles, red line indicated high traffic and blue line indicated mild traffic.

5. ANALYSIS

The analysis was carried out with images from different camera of four locations placed at different angles at road. The camera on top of road produced better result than that with camera at side of road.

As given in table 1, the number of vehicles detected by the system was compared against the actual number of vehicles in image frames. The result was 70% accurate for the camera at side of road and 90% accurate for the camera on top of road.

Table 1: Vehicle Detection

Location	Camera Position at Road	Actual Moving Objects	Detected Moving Objects	Accuracy %
Bhadrakali	Top	275	257	93.45
Sundhara	Top	173	152	87.86
Sital Niwas	Side	160	115	71.85
Baneshwor	Side	121	90	74.38

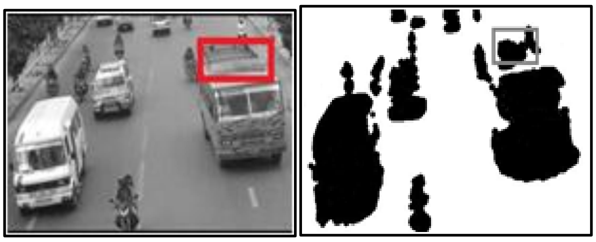

Figure 11: Large vehicle with rectangular shape on rooftop

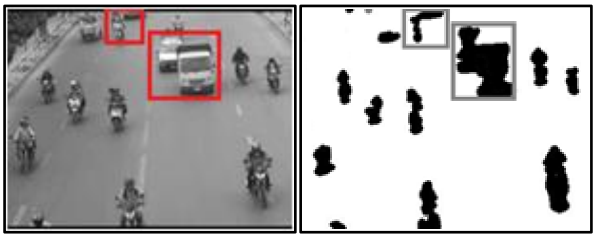

Figure 12: Vehicle occlusion, vehicle overlapping

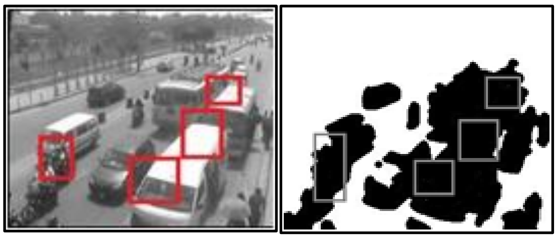

Figure 13: Vehicle occlusion for camera at side of road

The cent percent accuracy was not obtained because of various reasons. Even though the shadow of the vehicles didn't pose a problem, shapes of vehicles did account for incorrect count. As in figure 11, the long bus with rooftop was counted as two vehicles instead of one. Also as in figure 12, the vehicles that appeared overlapped together was counted as one instead of two. This problem was apparent for the camera at side of the road shown in figure 13. Here multiple numbers of vehicles appearing overlapped as seen from side of the road is detected as single object; hence the count is much less than the actual number. However vehicle overlapping was less for the camera at the top of the road compared to camera at side of road. Hence the result for camera on top of road was better compared to camera at side of road.

The road traffic congestion information of streets was displayed on map with different color lines to represent different traffic state of road as in figure 10. The lines of different colors were drawn on Openstreet map by defining array of coordinates i.e. latitude, longitude of the streets. Had Keyhole Markup Language (KML) street data of Kathmandu been available, it would have been easier to draw smoother lines on Openstreet Map.

6. CONCLUSION

To improve quality of transport management, reliable automatic vehicle detection system is necessary. Most of the existing surveillance cameras are positioned at side of road. The result of this project for that case was poor. However the camera positioned at top of road gave better coverage of the street so the system produced better result in this case. The system couldn't produce accurate result because of factors like overlapping of vehicles, shape of vehicles, presence of pedestrians, illumination, reflection from glass panels, etc.

Further improvement of the project is necessary to make the system robust. The static background subtraction technique produces inefficient result against illumination changes and weather changes. Automatic background extraction techniques can be implemented to address this problem. Also robust algorithm can be brought in order to address the light reflection. Result from side camera along with top camera can be used to address vehicle occlusions and vehicle overlapping.

With enhancements to address problems of illumination changes, weather changes and repositioning cameras at side of road to top of road to overcome problem of vehicle overlapping, this idea can certainly be implemented in cities like Kathmandu acting as reliable source of information about traffic congestion state of road to people travelling around the city. The information of road traffic density can be further used to automatically control traffic light there by replacing manual road traffic management by an automatic system.

7. ACKNOWLEDGMENTS

The authors gratefully acknowledge helpful discussion with C. McLean and B. Sthapit.

8. REFERENCES

[1] T. H. Thi, "A Robust Traffic Surveillance System for Detecting and Tracking Vehicles at Nighttime", University of Technology, Sydney, Faculty of Engineering, November, 2007.

[2] L. Vibha, M. Venkatesha, G. P. Prasanth, N. Suhas, P. D. Shenoy,K. R. Venugopal and L. M. Patnaik, "Moving Vehicle Identification using Background Registration Technique for Traffic Surveillance", Proceedings of the International MultiConference of Engineers and Computer Scientists 2008, Hong Kong, Vol. I IMECS 2008, 19-21 March, 2008.

[3] I. Jain and B. Rani, "Vehicle Detection Using Imge Processing and Fuzzy logic", International Journal of Computer Science and Communication, Vol. 1, no. 2, 2010, pp 255-257.

[4] D. J. Dailey, F. Cathey and S. Pumrin, "An Algorithm to Estimate Mean Traffic Speed Using Uncalibrated Cameras", In IEEE Transactions on Intelligent Ttransportations Systems, Vol. 1, no. 2, June, 2000, pp. 98-107.

[5] P. Srinivas, Y. L. Malathilatha and M. V. N. K. Prasad, "Image Processing Edge Detection Technique used for

Traffic Control Problem", International Journal of Computer Science and Information Technologies, Vol. 4 (1) , 2013, pp 17 – 20.

[6] A. J. Lipton, H. Fujiyoshi and R. S. Patil, "Moving Target Classification and Tracking from Real-time Video", In Proceedings of IEEE Workshop Applications of Computer Vision, 1998, pp. 8–14.

[7] R. C. Gonzalez and R. E. Woods, "Digital Image Processing", Second Edition, Prentice Hall, pp 123-124 and pp 596.

[8] H. E. Burdick, "Digital Imaging: Theory and Applications", McGraw-Hill, 1997, pp 423.

[9] T. Wood, "OpenLayers Workshop", pp 25.

[10] E. Hazzard, "OpenLayers 2.10 Beginner's Guide", Birmingham: Packt Pub., 2011, pp 104-118.