IOSR Journal of Mechanical and Civil Engineering (IOSR-JMCE)
e-ISSN: 2278-1684, p-ISSN: 2320-334X
PP 07-13
www.iosrjournals.org

Controlling Traffic Jam using Feature Detection and Object Detection Technique

N.R.Vikram[1], S.Sivaraman[2], R.Balamurugan[3]
CSE Department, Anna University, Paavai Engineering College (PEC), Namakkal, Tamilnadu, India

Abstract**:** *An efficient traffic management system is the frequent traffic jam at major junctions call. Bad traffic management leads to wastage of time, man-hours and increase in pollution levels that can be avoided on a city-wide scale by this system. This paper proposes to implement an intelligent traffic controller using real time image processing by developing self-adaptive system. Feature (Edge) detection is an important technique in many image processing applications such as pattern recognition, object recognition, medical image processing etc… The image sequences from a camera are analyzed using 2D edge detection and object counting methods to obtain the most efficient technique. This was done under MATLAB. By using, the obtained information the development of an android application can be done, by which the user can measure the traffic density at a particular location of choice.*

Keywords: *Traffic Management, Background subtraction, Edge Detection, Image Processing, Traffic Density.*

I. INTRODUCTION

Edge detection is the process of identifying and locating sharp discontinuities in an image.The discontinuities are abrupt changes in pixel intensity which characterizes boundaries of objects in a scene [6]. Edge detection is one of the most frequently used Techniques in digital image processing [7]. The boundaries of object surfaces in a scene often lead to oriented localized changes in intensity of an image called edges [8]. Classical methods of edge detection involves convolving the image with an operator (2-D filter) which is constructed to be sensitive to large gradients in the image while returning values of zero in uniform regions. Edge detection technique also transforms images benefiting from changes in grey tones in the image. Edges are signs of lack of continuity and ending as a result of this formation the edge is obtained without encountering any changes in physical qualities of image [4], [5]. There are extremely large numbers of edge detection operators available each designed to be sensitive to certain edges [8].Edge detection is difficult to implement in noisy images, since both noise and edges contains high frequency content [6]. Edge detection operator needs to be chosen to be responsive to gradual change which results from refraction or poor focus of the object with boundaries. This prevents problems of false edge detection, missing true edges, edge localization, and high computational time. In this paper we discuss a feature-based tracking system for detecting vehicles under various challenging conditions. Instead of tracking entire vehicles, features of vehicle are tracked to make system easy and effective. A common motion constraint is used as a salient features to make the system function fully even the lighting changes. The system works on latest technologies like digital image processing. The system consists of cameras that are fixed in lanes which are used to maintain and monitor traffic jam by capturing videos continuously. At particular time intervals frames are extracted, consecutive frames are compared and traffic jam is determined based on some parameters. The system is flexible, reliable and cost-effective. From the obtained information we can develop an android application by which the user will know the density of traffic at the destination. By using this application the user can save time by selecting the alternate route to reach destination. User opens the application and gets the list of locations present in the database. The user selects the location of his choice and density of traffic is returned.

II. PROPOSED SYSTEM

2.1 System Architecture

The various steps of our proposed system are described as follows: The camera is fixed on tolls or very tall structured buildings for capturing the vehicles that are passing in that signal for identifying the traffic density in that particular position. The videos are viewed for every three minutes for updating the traffic density to the user through android application.

2.2 Background Subtraction

In order to avoid the noise in the video we are using the background subtraction for the process of extracting the objects. Static background subtraction has been the traditional method for real - time segmentation of an object in video based system. Video- based techniques for outdoor environments are easily influenced by factors such as weather, change in illumination and motion. Hence, a static background proves insufficient and a robust background model is necessary to deal with change of luminance. We propose the use of the adaptive background technique as described in [9]. Generating the current background image based on segmentation results extracted from differencing the image with the previous extracted background is the basic idea of our method. The updated background (Bnew) is computed as a function of current background (Bo) and current frame I through the equation:

Fig.1.a Color image from real time video, 1.b after background Subtraction

2.3 Feature Detection

The foreground objects are extracted by subtracting the background. By using the edge detection algorithm the foreground objects are extracted from the real-time video. There is different edge detection techniques are available [10]. In this paper we are using the canny edge detector algorithm. This algorithm is now used widely in all the industry. When compared with the other entire edge detection algorithms, the accuracy and efficiency of this is more. It works by first smoothing the image and finds the image gradient to highlight regions with high spatial derivatives. It then tracks along these regions to suppress any pixel that is not at the maximum. Finally, through hysteresis, it uses two thresholds and if the magnitude is below the first threshold, it is set to zero. If the magnitude is above the high threshold, it is made an edge and if the magnitude is between the two thresholds, it is set to zero unless there is a path from this pixel to pixel with a gradient above the second threshold. This is to say that the two thresholds are used to detect strong and weak edges, and include the weak edges in the output only if they are connected to strong edges. The Canny edge detector is shown in Fig.2 which is widely considered to be the standard edge detection algorithm in the industry. It was first created by John Canny for his Master's thesis at MIT in 1983 [11], and still outperforms many of the newer algorithms that have been developed. Canny saw the edge detection problem as a signal processing optimization problem, so he developed an objective function to be optimized [11]. The solution to this problem was a rather complex exponential function, but Canny found several ways to approximate and optimize the edge-searching problem. The steps in the canny edge detector are as follows:

1. Smooth the image with a two dimensional Gaussian. In most cases the computation of a two dimensional Gaussian is costly, so it is approximated by two one dimensional Gaussians, one in the x direction and the other in the y direction.
2. Take the gradient of the image. This shows changes in intensity, which indicates the presence of edges. This actually gives two results, the gradient in the x direction and the gradient in the y direction.
3. Non-maximal suppression. Edges will occur at points the where the gradient is at a maximum. Therefore, all points not at a maximum should be suppressed. In order to do this, the magnitude and direction of the gradient is computed at each pixel. Then for each pixel check if the magnitude of the gradient is greater at one pixel's distance away in either the positive or the negative direction perpendicular to the gradient. If the pixel is not greater than both, suppress it.
4. Edge thresholding. The method of thresholding used by the Canny Edge Detector is referred to as "hysteresis". It makes use of both a high threshold and a low threshold. If a pixel has a value above the high threshold, it is set as an edge pixel. If a pixel has a value above the low threshold and is the neighbor of an edge pixel, it is set as an edge pixel as well. If a pixel has a value above the low threshold but is not the

neighbor of an edge pixel, it is not set as an edge pixel. If a pixel has a value below the low threshold, it is never set as an edge pixel.

Fig. 2 Canny applied to real time scene

2.4 Contour Tracing

After the edges are found the number of objects is to be counted as defined by the edges. For object detection and contour tracing there are many algorithms. These include the commonly used Radial Sweep method, Theo Pavlidis" Algorithm and Square Tracing Algorithm. However, we have implemented the Moore-neighborhood algorithm based on a similar method as in [12].

Fig 3.Counting no. of cars using Moore-neighborhood algorithm

The algorithm starts by choosing a random start point. When the current pixel is black, the Moore-neighborhood of is examined in clockwise direction starting with the pixel from which was entered and advancing pixel-by- pixel until a new black pixel in is encountered. The algorithm terminates when the start pixel is visited for the second time. The black pixel walked over will be the contour of the pattern. The efficiency of the algorithm improves greatly when we stop only after entering the start pixel in the same manner as entered initially. This is known as Jacob's stopping criteria. We have implemented this algorithm which does a decent job of identifying the number of cars in a given picture. The contour tracing algorithm enables us to define the boundary of the object as well as their size. We specify different size ranges to classify the various types of vehicles. This gives us a measure of the traffic density on each road at the intersection (refer Fig. 3). The traffic light is then regulated by allotting variable time according to the measured density and size of the vehicles.

2.5 Traffic Density Calculation

The edges that are detected are the features. These features are extracted from the image and it is labeled. By labeling it is very easy to identify the model of object. Depending upon the model, the traffic density at a particular position can be identified more accurately. Here we are using Kadir and Brady algorithm. When compared to all the other algorithms this works more effectively by using the two key aspects: entropy and scale. Kadir and Brady's algorithm is very general in speaking of a "local descriptor". As mentioned before, a possible and simple local descriptor is color intensity. All experiments in [13] are based on intensities with good results.

However, they point out that replacing the local descriptor spectral and wavelet entropy are discovered within this method. Clearly the sort of descriptor chosen also determines what features are salient. Regions of the image that can be modeled well (short) with the descriptor used are less salient because the complexity of the description is low. Based on another descriptor it may well be very complicated to model the same region and the saliency for that descriptor would then be high. This property means that the algorithm is a very general approach and modeling a good descriptor is an essential part of the problem. Finally a part of the problem, namely the selection of the “best” features has not been covered in this paper. Simple Thresholding works and the images in Fig. 4 have been created based on Thresholding. More careful models for the selection, such as clustering may of course improve results and prevent two similar features that are close to one another to be detected both. In the figure this can be seen by the many overlapping circles that represent the same object as different features.

Fig. 4 Result of the algorithm (most salient features selected by threshold).

2.6 Object Classification

In our captured traffic scenes, moving objects are typically vehicles or pedestrians. We use the ratio of height/width of each bounding box to separate pedestrians and vehicles. For a vehicle, this value should be less than 1.0, for a pedestrian this value should be greater than 1.5. But we also have to provide flexibility for special situations such as a running person, a long or taller vehicle. If the ratio is between 1.0-1.5, then we use the information from the corner list of this object to classify it as a vehicle or a pedestrian (a vehicle produces more corners). This is a simple way to classify moving objects into these two categories.

Fig 5. Left: An enlarged picture showing detected corners of vehicles marked by white dots. Right: Bounding boxes of moving vehicles and their centers marked by white crosses.

For moving object tracking we use a hybrid method based on bounding box and feature tracking. During the initialization period a data record is generated for each object: a label for indexing and the five elements of its vector. New positions are predicted using a Kalman filter. For each new frame, the predicted position is searched to see whether it can find any match with the previous data record. If a matching object region is found, it is marked as 'successfully tracked' and belongs to a normal move; if we cannot find any match, then the object may have changed lanes, or stopped, or exceeded the expected speed. So an unmatched object will be checked against already existing objects in the data record. If matched, then it is also marked as 'successfully tracked'; if still not yet matched, it will be marked as a new object and added to the data record. If an existing object is not being tracking for 5 frames, it will be marked as 'stopped'. According to the video capturing speed, we also define a threshold, which is used for marking 'tracking finished'. Matching is performed within certain thresholds for the different feature vector elements. The three main elements used for matching are: same color, a linear change in size, and a constant angle between the line 'corner point-upper left point' versus the line 'corner point-lower bottom point'. Occlusions are reported if bounding boxes are overlapping. In case of partial occlusions, calculated corners and further feature vector elements are tested for making a decision. Finally the data record will be updated using the results of the matching process

III. RESULTS AND DISCUSSIONS

To compare between various types of edge detection algorithms we tested their performance for ten images taken from real traffic intersections. After finding the edges, the picture was subjected to an object counting algorithm. The performance of the edge detector algorithms was defined by the number of vehicles accurately detected. The results are shown in Table I. Canny Edge detector was found to be the best among those compared (93.47%).The Boolean edge detector performs a decent job of marking the locations of edges, however it failed to complete the edges making object detection difficult. The Sobel and Prewitt operators are more adept at recognizing edges that are horizontal or vertical and are susceptible to noise (refer Fig 5), as also found in [14]. The Marr-Hildreth was found to be the most susceptible to noise and gave a lot of false results. The use of two thresholds by canny edge detector makes it less likely to be fooled by noise, and more likely to detect true weak edges, providing a better and fairly noise resistant method for the detection of edges. Hence we have used this method of detection in the paper, along with Moore- neighborhood method to count the vehicles marking the final step of our system.

Table 1: comparison of edge detection technique

Image	Actual no.	Marr	Boolean	Euclidean	Sobel	Canny
1	4	2	6	2	2	4
2	3	0	4	1	1	2
3	4	2	3	2	3	4
4	6	2	3	2	3	6
5	5	2	3	3	3	5
6	7	3	5	3	2	6
7	4	1	5	1	1	4
8	5	2	5	3	2	5
9	3	0	3	0	1	2
10	6	4	3	2	3	6
Accuracy		39.13	84.78	41.30	45.6	93.47

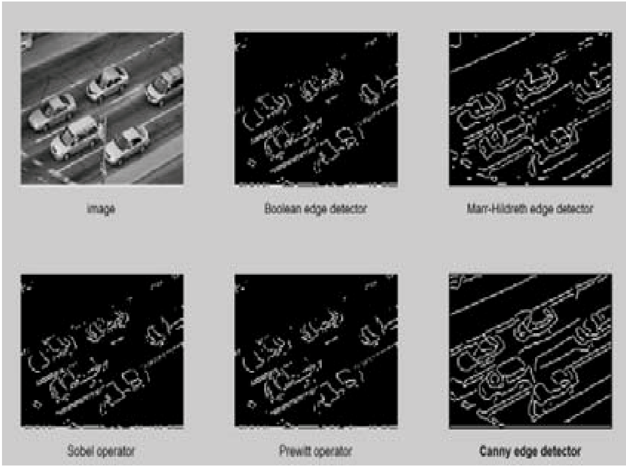

Fig 6. Comparison of various edge detection techniques.

Fig. 7: (a) Real-time image (b) Background image (c) Subtracted image (d) No. of vehicles = 3

The proposed system is used to analyze a real time traffic scene for a road (Fig 6 (a)). The adaptive background, updated from the scenes is shown in Fig. 6(b). The subtracted image then contains only the foreground objects (vehicles) as seen in Fig 6(c). Using image processing algorithms (Fig 6(d)), the number of vehicles in the lane can be found out. In this case, the number of vehicles is 3.

IV. CONCLUSIONS AND FUTURE WORK

In this paper we have successfully implemented an algorithm for a real-time image processing based traffic controller. Upon comparison of various edge detection algorithms, it was inferred that Canny Edge Detector technique is the most efficient one. Analysis of various contour tracing and object counting methods revealed the Moore neighborhood technique to be more robust when compared to the others. The paper demonstrates that image processing is a far more efficient method of traffic control as compared to traditional techniques. The focus shall be to implement the controller using DSP as it can avoid heavy investment in industrial control compute while obtaining improved computational power and optimized system structure. The hardware implementation would enable the project to be used in real-time practical conditions. More information about this method can be found in [14]. In addition, we propose a system to identify the vehicles as they pass by, giving preference to emergency vehicles and assisting in surveillance on a large scale.

REFERENCES

[1] P. Srinivas et al, Image Processing Edge Detection Technique used for Traffic Control Problem, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 4 (1),2013, 17 – 20

[2] Prof. Uma Nagaraj, Jinendra Rathod, Prachi Patil, Sayali Thakur, Utsav Sharma Traffic Jam Detection Using Image Processing, / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-962 Vol. 3, Issue 2, March - April 2013, pp.1087-1091

[3] Ireyuwa. E. Igbinosa, Comparison of Edge Detection Technique in Image Processing Techniques, Information Technology & Electrical Engineering, Volume 2, Issue 1 February 2013

[4] M.Ibrahiem, M.EL Emery, On the application of Artificial Neural in analysing and classifying human chromosome, Journal of Computer science vol. 2(1) pp.72-75 2006

[5] N.Senthilkumaran and Rajesh, A study on edge detection methods for image segmentation, Proceedings of the international Conference on Mathematics and Computer Science (ICMCS-2009) vol. 1, pp. 255-259, 2009

[6] R.Maini, H.Aggarwal. Study and comparison of various image edge detection techniques, International Journal of Image processing (IJIP), volume (3), issue (1) 2009.

[7] N.Senthilkumaran and Rajesh, Edge Detection Techniques for image segmentation-A survey, Proceedings of the international conference on managing next generation software applications (MNGSA-08) pp.749-760, 2008.

[8] N.Senthilkumaran and Rajesh, Edge detection Techniques for image segmentation-A survey of soft computing approaches, International Journal of Recent trends in Engineering, vol. 1 no 2, 2009.

[9] Jinglei, L. Zhengguang, and T. Univ, "A vision-based road surveillance system using improved background subtraction and region growing approaches," Eighth ACIS Int. Conf. On Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, Qingdao, August 2007, pp. 819- 822.

[10] M. Siyal, and J. Ahmed, "A novel morphological edge detection and window based approach for real-time road data control and management," Fifth IEEE Int. Conference Information, Communications and signal Processing, Bangkok, July 2005, pp. 324-328.

[11] R. Owens, "Lecture 6", Computer Vision IT412, 10/29/1997.http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/OWENS/LECT6/node2.html.

[12] K. Wang, Z. Li, Q. Yao, W. Huang, and F. Wang, "An automated vehicle counting system for traffic surveillance", IEEE In Conference. On Vehicular Electronics and Safety, Japan, Dec2007, pp. 1-6.

[13] Saliency, Scale and Image Description, T. Kadir & M. Brady, Robotics Research Laboratory, Department of Engineering Science, University of Oxford.

[14] M. Juneja, P. Sandhu, "Performance evaluation of edge detection techniques for images in spatial domain", Int. Journal of Computer Theory and Engineering, Singapore, vol-1, no. 5, Dec 2009, pp. 1793-8201.