OBJECT DETECTION AND TRACKING: EXPERIENCES WITH CONVENTIONAL IMAGE PROCESSING TECHNIQUES ON VEHICULAR TRAFFIC VIDEO

[1]RASHMIT SINGH GIRVAR SINGH SUKHMANI, [2]PRASUN KUMAR GUPTA, [3]SWAPNIL JHA

[1,2]Geoinformatics Department, Indian Institute of Remote Sensing, Kalidas Road, Dehradun, India
[3]BIT Mesra
E-mail: [1]rashmit@iirs.gov.in, [2]prasun@iirs.gov.in, [3]swapnil1270.11@bitmesra.ac.in

Abstract- Much work has been done in developing image processing techniques for data mining purposes; either by image enhancement or feature extraction. Seldom have these methods been applied to solve current day problems in transport management. This paper emphasis on the use of well established techniques (mountain gap & filters) in modified formats to achieve vehicle detection and tracking.

Keywords- Image processing, Object Detection, Object Tracking, Filtering.

I. INTRODUCTION

Visual tracking has emerged as an important component of systems in several application areas including visual-based control surveillance medical imaging and visual reconstruction. The problem poses a challenge because of the multi-variate and multi-dimensional inputs in various application domains. The central challenge in visual tracking is to determine the image configuration of the target region of an object as it moves through a camera's field of view. As object tracking has a lot of applications, many algorithms have been proposed to solve the problem. Changing illumination, scene changes and shadows are typical problems which make the problem challenging. A surveillance application usually consists of some sort of moving object detection, object tracking and higher order processing. A lot of existing methods first perform computationally expensive spatial segmentation based on gray scale values. This is not necessary in lots of applications, where only moving objects need to be tracked. This paper mainly concentrates on the two levels, moving object detection and object tracking. In this paper we propose an effective and efficient method for tracking moving objects in video sequences. In this paper we are not considering the camera motion, as our camera is stationary with small field of view.

II. BACKGROUND

Before introducing our algorithm, we want to give a short review of relevant work to establish the necessary background.

A. Filtering – Smooth Filter (Average Box)
Removal of noise in the image is a vital processing task in image processing techniques. The important property of good image de-noising model is that it should completely remove noise as far as possible as well as preserve edges. Basically, there are two models .i.e. linear and non linear models. The benefit of linear noise removing model is the speed but the problem is that it does not preserve edges as non linear noise removing model does. The technique of smoothing the image is also known as blurring. This smooth filtering function returns a copy of array smoothed with a boxcar average of the specified width. The result has the same type and dimensions as the array. The algorithm used here is given in Eq. 1.

$$R = \frac{1}{w}\sum_{j=0}^{w-1} A_{i+j-\frac{w}{2}}$$
$$\text{where,} \quad i = \frac{w-1}{2}, \ldots\ldots N - \frac{w+1}{2} \tag{1}$$

'R' is the final output of the function, 'w' is the window size, 'A' is the array on which the smoothing window is made to run and 'N' stands for the number of elements in the array of the image.

It is generally an average filter which is useful for removing grain noise from the photographs. Because each pixel gets set to the average of the pixels in its neighbourhood, local variations caused by grain are reduced. This type of filter is a linear filter which uses a mask over each pixel. Each component of the pixels which fall under the mask are averaged together to form a single pixel.

B. Filtering – Mode Filter
In this type of filter, each pixel value is replaced by its most common neighbour. This is a particularly useful filter for classification procedures, where each pixel corresponds to an object which must be placed into a class. It is a non linear moving window filter which replaces the centre value of the window with the maximum times occurred value in the mask. An

example of this windowing method is shown in Fig. 1.

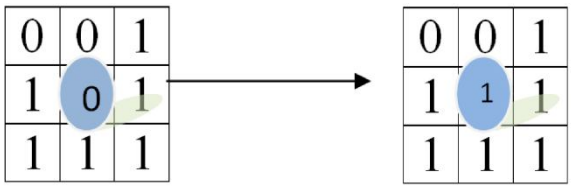

Fig. 1 Working model of Mode Filter

C. Thresholding

Segmentation problems requiring multiple thresholds are best solved using region growing methods. Thresholding can be denoted as Eq. 2.

$$T = T[x, y, p(x,y), f(x,y)] \tag{2}$$

where f(x,y) is gray- level at (x,y) and p(x,y) denotes some local property, for example average gray level in neighborhood. A thresholded image g(x,y) is defined as Eq. 3.

$$g(x,y) = \{1, f(x,y) > T\}$$

$$g(x,y) = \{0, f(x,y) \leq T\}$$

$$where, 1 \text{ is object and } 0 \text{ is background.}$$

$$When\ T = T[f(x,y)], threshold\ is\ global,$$

$$When\ T = T[p(x,y), f(x,y)], threshold\ is\ local,$$

$$When\ T = T[x, y, p(x,y), f(x,y)],$$

$$threshold\ is\ dynamic\ or\ adaptive. \tag{3}$$

This method is used to convert the images into the binary images where 1 corresponds to the objects and 0 corresponds to the background. In the frame difference thresholding is explained using change detection mask technique for segmentation. The generated masks are further processed to obtain final object masks. The processing speed of the algorithm is high compared to other thresholding algorithms.

In our case we have improved upon this technique and used it for object detection. The algorithm in uses successive frames algorithm based on to detect changes in the object as shown in Eq. 4 and Eq. 5.

$$FD(x,y,t) = |I(x,y,t) - I(x,y,t-1)| \tag{4}$$

$$FDM(x,y,t) = 1\ if\ FD > Th$$

$$FDM(x,y,t) = 0\ if\ FD \leq Th \tag{5}$$

where I is the DN value in frame data, FD is frame difference between two consecutive frames, and FDM is the generated Frame Difference Mask. Pixels belonging to FDM are moving pixels inside the frames. The parameter 'Th' is threshold value of Digital Number (DN) which has to be set in advance before starting of the algorithm. In our case, I(x,y,t-1) is replaced by Irf (x,y,t1) in all the frames. Irf is the background reference frame which has been taken at some arbitrary time t1. So, now the algorithm changes can be denoted as Eq. 6 and Eq. 7.

$$FD(x,y,t) = |I(x,y,t) - I_{rf}(x,y,t_1)| \tag{6}$$

$$FDM(x,y,t) = 1\ if\ FD > Th$$

$$FDM(x,y,t) = 0\ if\ FD \leq Th \tag{7}$$

D. Correlation Coefficient

Spearman's (ρ) rank correlation of two sample populations is used to track objects in different frame along with its significance values. The value of Spearman's rank correlation, as given in Eq. 8, lies between [-1,1] where -1 represents perfect negative correlation, 1 represents perfect positive correlation and '0' represents no correlation between the data. But practically values are not integers, so closer to 1 is more agreement between both the data and closer to -1 means more disagreement between the data.

$$\rho = \frac{\sum_{i=0}^{N-1}\left(R_{x_i} - \overline{R_x}\right)\left(R_{y_i} - \overline{R_y}\right)}{\sqrt{\sum_{i=0}^{N-1}\left(R_{x_i} - \overline{R_x}\right)^2}\sqrt{\sum_{i=0}^{N-1}\left(R_{y_i} - \overline{R_y}\right)^2}} \tag{8}$$

It is a non parametric (distribution- free) rank statistic as it is a measure of strength of the associations between two variables. The Spearman's rank correlation is a measure of monotone association that is used when the distribution of the data make Pearson's correlation coefficient undesirable or misleading. The range of significance values is in between. Smaller the significance value indicates significant correlation.

III. METHODOLOGY

A. Object detection

In this section, we introduce our algorithm for tracking moving objects in video sequences. A video sequence contains a series of frames. Each frame can be considered as an image. If an algorithm can track moving objects between two digital images, it should be able to track moving objects in a video sequence which is taken from still camera.

The algorithm starts with two frames, one is the input frame (Fig. 2 – a) and the other one is the background reference frame (Fig. 6). Here background frame is the image where there are no objects except background. This image is one of the chosen frames of video sequence. Background reference frame is subtracted from the input frame as a result we got

only objects in the output image (Fig. 2 – b). Smoothing filter was applied for noise suppression in order to preserve the

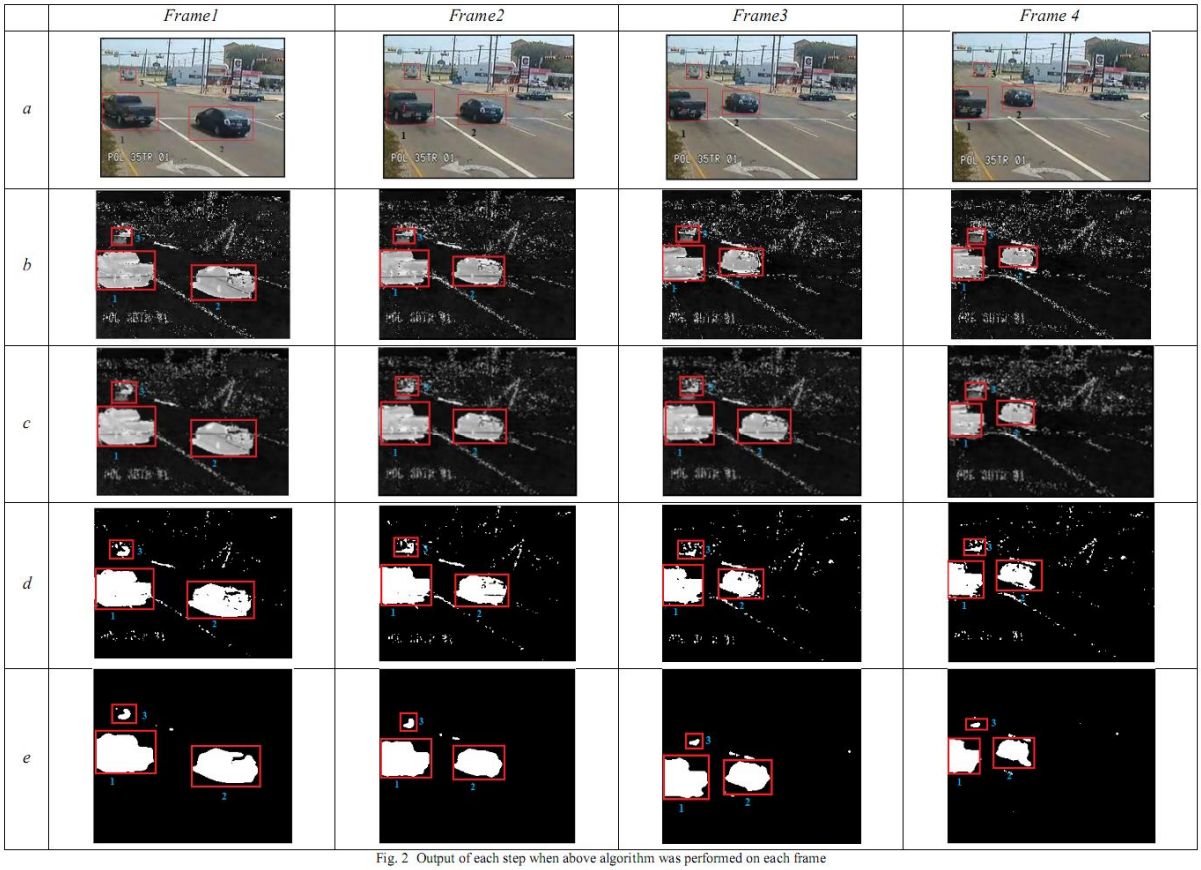

Fig. 2 Output of each step when above algorithm was performed on each frame

spatial frequency details in an image (Fig. 2 – c). It has helped us in removing noise spikes i.e. isolated pixels of exceptionally low or high pixel intensity. As vehicles being metallic bodies the difference of DN values at the location of moving objects will be larger than the background. To enhance these differences in an image, thresholding was done (Fig. 2 – d). After thresholding the image, image got converted into binary where object locations are having value 1 whereas background as 0. For filling up the gaps in object locations with value 1 we have used moving window mode filter (Fig. 2 – e).

After completing the procedure above, we will have different size of blocks classified as objects. Some of the blocks are obtained because of moving objects, while others are just the result of noise. Therefore, we need to group the blocks of the same objects and eliminate the noise effect. Image is an array consisting of columns and rows. Each row values are added to get the column vector. A zero element in the column vector means that there are no object pixels in the corresponding row. If the number of consecutive zeros in larger than the preset threshold, it is called "Gap". The group of elements between "Gaps" is called the "Mountains" and the element with the largest value is called "Peak". If the width of a "Mountain" is larger than a preset threshold and its "Peak" is high enough, then it is concluded that there is at least one object in the "Mountain". If an image is taken as an m × n matrix, after the above procedure, all the possible objects are in q smaller matrices and each of them has n columns. If the same concept is applied to these images and values are added for each column instead of the row, matrices containing the object will be smaller in total size. This processing is done continuously till the total size of the matrices does not change any more. After the iteration, obtained matrices will be the object location in the frame. As a whole, every object location can be isolated by above mentioned methodology.

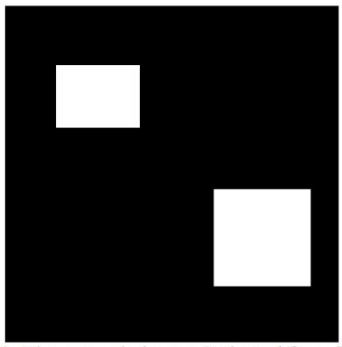

Fig. 3 The example image (Adapted from [22])

To explain the "Gap/Mountain" method described above more clearly, we will give a simple example. A

200 × 200 binary image is shown in Fig. 3. The gray scale value for the background is set to zero and there are two rectangular objects of gray-scale value set to one in the image. By adding the value of each row, we get a column vector of 200 elements. If we set thresholds for "Gap" and "Mountain" width to be 20, we can see that in Fig. 4 there are three "Gaps" and two "Mountains" and the "Mountains" are from column 32 to 92 and from column 130 to 170.

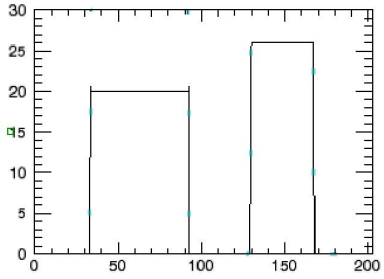

Fig. 4 The plot of column vector (x-axis denoting number of pixels and y-axis denoting column number)

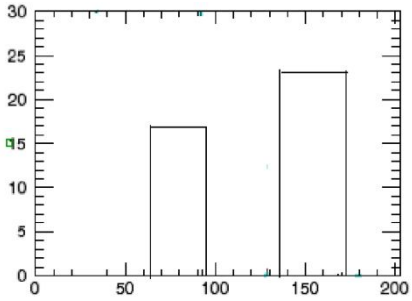

Fig. 5 The plot row vector (x-axis denoting number of pixels and y-axis denoting row number)

Fig. 6 Background reference frame

Hence, instead of knowing the possible objects are in the initial 200 × 200 matrix, now we know that they are present in two smaller matrices. First one is approximately in 60 by 200 and the other object is in 40 by 200. Then, similar idea is applied to these two smaller matrices one by one. Next step was to add all of the values of columns and 'Mountains' and 'Gaps' are detected in the corresponding row vectors. Similarly, in Fig. 5 mountains are from 62 to 92 and from 135 to 170. This algorithm is applied iteratively until the matrices do not change anymore. We are mainly interested in the centre coordinates of these mountains. In this example we got two mountains when added all rows to get column vector, so it will have 2 centre points which will corresponds to column vectors. Similarly, we got two centre points when we added all columns to get row vector. Now we have two points of column vector and two points of row vector. Now converting these points into image coordinate system we got 4 combinations of points. Then we searched around these points for the objects in searching window of 40×40 pixels. In this example, we got 4 combinations of coordinate .i.e. (77,62), (77,150), (150, 62) and (150,150). This object finding approach will be much easier for binary images because of this reason images are converted into binary. In this example, there are two objects one at (77,62) and other one is at (150, 150). In this we way are going to find the number of objects in each frame. This method of finding out the centroid of the objects is much easier and faster. Here we have assumed that the size of the objects will not change much between adjacent frames.

TABLE I
TABULAR ALGORITHM

1	Video fragmentation into Frames
2	Input Frames [IF] – Background Frame [BF]
3	Smooth Filtering [SF]
4	Thresholding [TH]
5	Mode Filtering [MF] (Gap filling technique in the objects)
6	Determining object areas
7	Determining the Centroid of the objects
8	Finding out DN (Digital Number) values around the Centroid
9	Tracking Objects in the current frame by "Current/Previous" And "Current/Next" comparisons using Spearman's correlation

B. Object tracking

Now the next step is to track these objects in different frames. The object locations determined in each frame as explained in section III A are used as a centroid of window size 50×50 pixels to create a subset in each input frame. For each subset in input frame are correlated with subsets present in previous 4 frames. The intensity based area correlation is applied for each subset. The subset in one frame having maximum correlation 'ρ' with a subset in another frame are considered to be of same object. Here we are using Spearman's (ρ) rank correlation of two sample subsets of DN values to track objects in different frames along with the significance values.

This algorithm will help us in finding out number of objects in each frame and also to track these objects in different frames. The steps involved in this algorithm are given in Table I.

IV. RESULTS

In this section, we will show the experimental results of our algorithm. The algorithm is tested on a vehicular traffic video. The results from proposed methodology are shown in Fig. 2; where the time interval between each adjacent frame is 2 seconds (denoted by columns Frame 1, 2, 3 and 4). Objects areas are highlighted by using special effects in each frame. The number of vehicles detected in each frame is given in Table II. The objects can be tracked based on the rank correlation and the final results are given in Table III.

TABLE II
NUMBER OF VEHICLE IN EACH FRAME

Frames	Number of Vehicles in frame moving	Number of vehicles in frame moving detected
Frame 1	3	3
Frame 2	3	3
Frame 3	3	3
Frame 4	3	3

As it can be seen in above tables that number of vehicles detected in each frame is correct when it was validated with actual video sequence. Objects were tracked in each frame by calculating Spearman's correlation coefficient of object areas DN in each frame. Object having maximum correlation between the two frames was verified to be the same.

TABLE III
TRACKING RESULT

Compared Frame	Maximum Rank Correlation
Frame 1 – Frame 2	
Object 1	0.47
Object 2	0.64
Object 3	0.67
Frame 1 – Frame 3	
Object 1	0.52
Object 2	0.49
Object 3	0.58
Frame 1 – Frame 4	
Object 1	0.55
Object 2	0.58
Object 3	0.48
Frame 2 – Frame 3	
Object 1	0.82
Object 2	0.60
Object 3	0.62
Frame 2 – Frame 4	
Object 1	0.62
Object 2	0.52
Object 3	0.49
Frame 3 – Frame 4	
Object 1	0.73
Object 2	0.61
Object 3	0.43

CONCLUSION

In this paper we have shown that two important steps of video surveillance namely object detection and object tracking can be combinedly done using traditional image segmentation methods with modification. We have used traditional techniques such as smoothening, thresholding and mountain gap to do object detection; which is very resource intensive operations when done separately for image segmentation. But by combining these, the computational cost required for next step of video surveillance .i.e. object tracking using Spearman's (ρ) coefficient method, is greatly reduced. In these frames, as the object moves away from the camera, the shape of the object decreases due to scale variation. In such case the area based cross correlation is less sensitive to actual DN value. In this context the selection of Spearman's (ρ) coefficient is taken as measure of correlation. It also shows that it is appropriate in the case of scale variation. The Spearman's (ρ) coefficient method for object tracking also yielded satisfactory results – giving vehicle tracking operations a significant boost. The developed methodology can hence be used as a fast and inexpensive model for vehicular traffic movement management applications.

Our experimental results show that the algorithm is efficient to eliminate illumination problem and can give more accurate results. But one can even develop algorithms to weigh in different tracking methods to achieve more accurate results. Since our algorithm could not completely solve the problem for occlusion (as seen in Table III), there is further scope to "occlusion killer algorithms".

REFERENCES

[1] P. K. Allen, A. Timcenko, B. Yoshimi, and P. Michelman, "Automated tracking and grasping of a moving object with a robotic hand-eye system," Robotics and Automation, IEEE Transactions on, vol. 9, no. 2, pp. 152–165, 1993.

[2] S. Hutchinson, G. D. Hager, and P. I. Corke, "A tutorial on visual servo control," Robotics and Automation, IEEE Transactions on, vol. 12, no. 5, pp. 651–670, 1996.

[3] N. P. Papanikolopoulos, P. K. Khosla, and T. Kanade, "Visual tracking of a moving target by a camera mounted on a robot: A combination of control and vision," Robotics and Automation, IEEE Transactions on, vol. 9, no. 1, pp. 14–35, 1993.

[4] E. D. Dickmanns and V. Graefe, "Dynamic monocular machine vision," Machine vision and Applications, vol. 1, no. 4, pp. 223–240, 1988.

[5] R. Howarth and H. Buxton, "Visual surveillance monitoring and watching," Computer Vision—ECCV'96, pp. 321–334, 1996.

[6] T. Frank, M. Haag, H. Kollnig, and H. H. Nagel, "Tracking of occluded vehicles in traffic scenes," Computer Vision—ECCV'96, pp. 485–494, 1996.

[7] E. Bardinet, L. Cohen, and N. Ayache, "Tracking medical 3D data with a deformable parametric model," Computer Vision—ECCV'96, pp. 315–328, 1996.

[8] P. Shi, G. Robinson, R. Todd Constable, A. Sinusas, and J. Duncan, "A model-based integrated approach to track myocardial deformation using displacement and velocity constraints," in Computer Vision, 1995. Proceedings., Fifth International Conference on, 1995, pp. 687–692.

[9] E. Boyer, "Object models from contour sequences," Computer Vision—ECCV'96, pp. 109–118, 1996.

[10] L. S. Shapiro, Affine analysis of image sequences. Cambridge University Press (New York), 1995.

[11] C. Tomasi and T. Kanade, "Shape and motion from image streams under orthography: a factorization method," International Journal of Computer Vision, vol. 9, no. 2, pp. 137–154, 1992.

[12] I. Haritaoglu, D. Harwood, and L. S. Davis, "W4: real-time surveillance of people and their activities," Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 22, no. 8, pp. 809–830, 2000.

[13] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers, "Wallflower: Principles and practice of background maintenance," in Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on, 1999, vol. 1, pp. 255–261.

[14] C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland, "Pfinder: Real-time tracking of the human body," Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 19, no. 7, pp. 780–785, 1997.

[15] G. Gupta, "Algorithm for Image Processing Using Improved Median Filter and Comparison of Mean, Median and Improved Median Filter," International Journal of Soft Computing, vol. 1, no. 5, 2011.

[16] H. C. Huang, C. M. Chen, S. D. Wang, and H. H. S. Lu, "Adaptive symmetric mean filter: a new noise-reduction approach based on the slope facet model," Applied optics, vol. 40, no. 29, pp. 5192–5205, 2001.

[17] P. Patidar, M. Gupta, S. Srivastava, and A. K. Nagawat, "Image De-noising by Various Filters for Different Noise," International Journal of Computer Applications IJCA, vol. 9, no. 4, pp. 24–28, 2010.

[18] R. J. Radke, S. Andra, O. Al-Kofahi, and B. Roysam, "Image change detection algorithms: a systematic survey," Image Processing, IEEE Transactions on, vol. 14, no. 3, pp. 294–307, 2005.

[19] S. Natarajan, "An efficient Video Segmentation Algorithm with Real time Adaptive Threshold Technique," algorithms, vol. 2, no. 4, 2009.

[20] J. J. Cai, J. M. Macpherson, G. Sella, and D. A. Petrov, "Pervasive hitchhiking at coding and regulatory sites in humans," PLoS genetics, vol. 5, no. 1, p. e1000336, 2009.

[21] E. Lehmann, "Nonparametrics: statistical methods based on ranks (POD)," Recherche, vol. 67, p. 02, 2006.

[22] Y. Wang, R. E. Van Dyck, and J. F. Doherty, "Tracking moving objects in video sequences," in Conference on Information Sciences and Systems, 2000, vol. 2, pp. 24–29.

★★★