HYBRID SALIENT MOTION DETECTION USING TEMPORAL DIFFERENCING AND KALMAN FILTER TRACKING WITH NON-STATIONARY CAMERA

Xuesong Le
School of ICT
Griffith University
Gold Coast, Australia
x.le@griffith.edu.au

Ruben Gonzalez
School of ICT
Griffith University
Gold Coast, Australia
r.gonzalez@griffith.edu.au

Abstract— **Uncertain motion of typical surveillance targets, e.g. slow moving or stopped, abrupt acceleration, and uniform motion makes a single salient motion detection algorithm unsuitable for accurate segmentation. It becomes even more challenging in case of the camera is non-stationary. In this paper, first, a simple local adaptive temporal differencing method is proposed to detect moving objects boundaries and partial interiors. To improve the accuracy of detection, a Bottom-up Variable Block Size block matching method is employed to identify the existence of possible moving object blocks and then an adaptive Kalman filter is applied to distinguish salient motions from other distracting motions. At last, the motion data from two algorithms are successfully fused to determine whether a region has been changed or not. Experimental results comparing the proposed and other competing methods are evaluated objectively and show that the proposed method achieves promising motion results for a variety of real environments.**

Keywords—temporal differencing; block matching; kalman filtering; non-stationary camera; salient motion detection

1. INTRODUCTION

Detecting salient motion, defined in [1] as motion from a typical surveillance target (a person or a vehicle) in image sequence is one of main tasks in some promising applications, such as video surveillance, traffic monitoring, etc. However, it is still in its early developmental stage and needs to improve its robustness depending on the specific scene conditions. Some of the most challenging problems are those in which motion is being exhibited not just by the objects of interest, but also by other factors such as varying illumination, dynamic backgrounds, crowded scenes and occlusions. In case of the camera is non-stationary, moving object detection problem becomes even more challenging, since the background is not static and background subtraction methods cannot be employed anymore.

Several techniques for moving object detection been presented recently for non-stationary cameras [2] [3]. Among them, temporal differencing is the simplest method to detect moving objects by thresholding the gray level difference image. Rosin [4] approximated the camera noise as a normal distribution $N(\hat{\mu}, \hat{\sigma})$ using simple robust statistics method and chose global threshold t as $\hat{\mu} + c\hat{\sigma}$, where c is a constant. Aach et al. [5] evaluated a set of grey level differences inside a small decision region, instead of a single difference to decide whether a pixel at location p is changed or not. Leng [6] accumulated frame differences of past N frames together to detect the regions with low motion or low texture. However, such accumulation expands the actual changed region when the object is moving fast. Temporal differencing-based methods tend to cause small holes and cannot detect the complete shape of a moving object.

Rather than temporal differencing, another strategy to find moving objects from consequent video frames is by using the optical flow of the frames. To separate flow vectors which represent the movement of objects from those of background, classification processes are needed [7] [8]. Other optical flow methods avoid the flow vectors classification by compensating the camera motion first and then detecting the moving object between the previous frame and the motion compensated current frame [2] [9]. The resulting optical flow pixels whose magnitudes of optical flow vectors greater than threshold, T, will be considered as moving pixels. Tian [10] detects salient motion in complex environments by combining temporal difference imaging and temporal filtered optical flow based on the assumption that the object moves in a consistent direction for a period of time. The major drawbacks of using optical flow include: it is sensitive to light sources; how to choose threshold, T, remains a difficult problem; the computational cost of the approach is very expensive since it is a pixel by pixel processing approach, which makes it very difficult to apply in a real-time system. The alternative solution is to adopt batch processing approaches; such as block matching (BM). BM based motion detection algorithm divides compensated current frame into a matrix of macro blocks and then each macro block is compared with the corresponding block and its adjacent neighbours in the previous frame. BM is computationally more economical compared to the optical flow technique [11] for motion estimation and compensation. However, in practice, an object is defined by a contiguous set of pixels not a rectangular window. Block-based motion detector lacks the power to exactly localize the object: the entire pixels inside the best match block are classified as foreground.

A single salient motion detection algorithm is sensitive to illumination changes, compression artifacts, a noisy environment, or camera displacement, which are likely to generate numerous false positives. False negatives can also be induced by non-uniform motion of surveillance targets, e.g. slow moving or stopped, abrupt acceleration. With such scenarios, we examine the feasibility of using the temporal differencing in conjunction with Kalman filtering based BM algorithm for motion detection with a non-stationary camera. This paper starts with a simple local adaptive temporal differencing method to detect moving objects boundaries and partial interiors in section 2. To improve the detection of the motion, a Bottom-up Variable Block Size BM method is employed in section 3.1 to identify the existence of possible moving object blocks first. Then in section 3.2, an adaptive Kalman filter is applied on these identified motion blocks to distinguish salient motions from other distracting motions. In section 4, first, an efficient colour image segmentation algorithm is applied to group pixels in the image into coherent atomic regions. Then the motion activity in each segmented region is calculated based on the results of two motion detection algorithms respectively. Finally, the motion data from two algorithms are successfully fused to decide whether a region has been changed or not. In section 5, we successfully apply this algorithm to various real video sequences in comparison to several standard motion detection algorithms, where the issues of false positives and negatives, robustness to noise presence are addressed. Section 6 draws a conclusion.

2. TEMPORAL DIFFERENCING

In this section, we use Rosin's method [4] to approximate the camera noise distribution $N(\mu, \sigma^2)$ first. We start by computing the grey level difference image $d(p)$ between the two consecutive frames. The index p denotes the pixel locations on the difference image grid. If less than half of the gray level difference image, $d(p)$ is in motion, the camera noise can be modelled by a normal distribution $N(\mu, \sigma^2)$. The population mean μ and standard deviation σ can be estimated using simple robust statistics method, median of all absolute deviations from the median, which does not require any distribution assumptions and its breakdown value is 50%.

$$\text{MAD} = \text{median}(|d(p) - \text{median}(d(p))|), \tag{1}$$

$$\hat{\mu} = \text{median}(d(p))|), \text{and } \hat{\sigma} = 1.4826 \times \text{MAD} \tag{2}$$

where 1.4826 is a normalisation factor wrt. a normal distribution as defined in [12].

Then we follow Aach's [13] method to decide whether pixel at location p is changed or not inside a small decision window w. At first, each pixel in a window, w is normalised as follows:

$$n_w(p) = (|d_w(p) - \hat{\mu}|/\hat{\sigma})^2 \tag{3}$$

where $\hat{\mu}$ and $\hat{\sigma}^2$ are the estimated mean and the estimated variance of the camera noise.

If there is no change occurred at position p, the normalised squared difference, $n_w(p)$ is typically assumed to be independent and identically distributed. Thus, the local sum $t_s = \sum_{p \in W} |d_k(p) - \hat{\mu}|^2 / \hat{\sigma}^2$ inside W obeys a chi-squared distribution with as many degrees of freedom as there are pixels inside W. Given an acceptable significance level, α, interpreted as the probability of rejecting H_0 although it is true, and N degrees of freedom, its corresponding threshold t_α can be computed from the χ^2 distribution table. The chi-squared statistic t_s is now tested with t_α at each location p on the difference image. The change mask, $B_{x \in W}(p)$ of pixel at location p inside window W is marked as changed whenever it exceeds t_α, otherwise as unchanged.

$$B_{x \in W}(p) = \begin{cases} 1 & if\ t_s > t_\alpha \\ 0 & otherwise \end{cases} \tag{4}$$

3. KALMAN FILTER BASED BLOCK MATCHING

The idea for Kalman filter-based block matching in motion detection is motivated by the fact that BM itself may not detect the "genuine" motion rather the position of a good match. BM algorithm does not work well for: video sequences with non-uniform motion content; there are occlusions among the objects, as well as disappearing of objects and appearing of new ones. To solve this problem, a Bottom-up Variable Block Size (BVBS) BM method is employed to identify the existence of possible moving object blocks first. Then an adaptive Kalman filtering is proposed to add robustness to the moving object detection by improving the reliability of correctly detected objects while reducing the noise presence at the same time.

3.1. Bottom-up Variable Block Size (BVBS) BM

In a typical block matching, the block size is critical for the performance of the motion estimation. High motion details can be detected by using small block size because pixels in a small block are more likely to share the same motion vector. But the resulting motion estimate is very sensitive to inter-frame noise. On the other hand, a large block size is less sensitive to noise, but detect less motion detail. A large search area can detect fast motion, but also requires a large computation time, and is also sensitive to noise. When the estimated motion is not the true physical motion, the best matching can create false trajectory.

To solve those problems, BUVBM is proposed. A small-size block is chosen at the beginning to detect a greater number of motion blocks, first. Then the blocks size is doubled in each dimension. A set of blocks $\{B_i, i = 1,2,3,4\}$ at smaller size which share the same parent block S_b at larger size are defined as follows:

$$S_b = B_1 \cup B_2 \cup B_3 \cup B_4, \ where\ B_{i,1 \le i \le 4} \tag{5}$$

where 1,2,3 and 4 correspond to the top left, top right, bottom left, bottom right children block of parent block S_b respectively.

BM is applied on S_b only if at least one of its children blocks, B_i detected as motion block at smaller size during the previous step. This BM process can be repeated as many times as desired or until the block size reaches the predefined maximum size. In such way, more false motion blocks can be discarded at larger block size while the true motion details found at smaller block size are well kept.

3.2. Object Block Tracking with Adaptive Kalman filter

The Kalman filter, governed by the linear stochastic difference equation, provides a recursive solution to predict a process's state, and using measurements to correct these predictions. The algorithm works in a two-step process, time update and measurement update.

3.2.1. Time Update

It begins each iteration by predicting the process's state, $\bar{x}_k$ at step k, given knowledge of previous estimate $\hat{\boldsymbol{x}}_{\boldsymbol{k-1}}$.

$$\bar{x}_k = A\hat{\boldsymbol{x}}_{\boldsymbol{k-1}} + B_k u_k \tag{6}$$

$$\bar{P}_k = A\hat{P}_{k-1}A^T + Q, \tag{7}$$

where $\bar{P}_k$ is priori estimate error covariance.

Since the motion movement of object block in X and Y coordinate is totally independent of each other, Kalman filtering can be applied in X and Y coordinate separately. In each coordinate, the dynamics of an object block is described by

$$x_k = x_{k-1} + \dot{x}_{k-1}\Delta k + \ddot{x}_{k-1}\frac{\Delta k^2}{2}, \tag{8}$$

$$\dot{x}_k = \dot{x}_{k-1} + \ddot{x}_{k-1}\Delta k\,, \tag{9}$$

$$\ddot{x}_k = \ddot{x}_{k-1} + a_k\,, \tag{10}$$

where x_k, $\dot{x}_k$, $\ddot{x}_k$, Δk are position, velocity, acceleration and the time difference at time k, respectively. Since we have no known control inputs, $B_k u_k$ item in equation 6 is discarded. The specific equations for the time updates in motion tracking are presented below:

$$\begin{bmatrix} x_k \\ \dot{x}_k \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} 1 & \Delta t & \frac{\Delta t^2}{2} \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{k-1} \\ \dot{x}_{k-1} \\ \ddot{x}_{k-1} \end{bmatrix}, \tag{11}$$

$$P_k^- = \begin{bmatrix} 1 & \Delta t & \frac{\Delta t^2}{2} \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} P_{k-1} \begin{bmatrix} 1 & 0 & 0 \\ \Delta t & 1 & 0 \\ \frac{\Delta t^2}{2} & 1 & 1 \end{bmatrix} + Q_k \tag{12}$$

The determination of the process noise covariance Q_k is generally more difficult as the direct observation of the estimating process is not always feasible. In this work, we model that the acceleration as a piecewise constant Wiener process [14], i.e., acceleration remains constant for the duration of each time period, but differs for each time period, and each of these is zero-mean white sequence.

$$Q_k = q \begin{bmatrix} \Delta t^5/20 & \Delta t^4/8 & \Delta t^3/6 \\ \Delta t^4/8 & \Delta t^3/3 & \Delta t^2/2 \\ \Delta t^3/6 & \Delta t^2/2 & \Delta t \end{bmatrix} \tag{13}$$

where q is the power spectral density, of the continuous-time white noise. For the simplicity of calculation, the state covariance matrix, P_0

is initialised to be a multiple c of the process noise covariance matrix $P_0 = cQ_k$ where c is a constant.

3.2.2. Measurement Update

The measurement, $\boldsymbol{z_k}$, of the state $\boldsymbol{x_k}$ can be represented by a linear equation of the form

$$\boldsymbol{z_k} = H_k \boldsymbol{x_k} + v_k \tag{14}$$

where $\boldsymbol{z_k}$ is the $m\times1$ measurement made at time k, $\boldsymbol{H_k}$ is the $m\times n$ observation matrix which relates the state to the measurement z_k and $\boldsymbol{v_k}$ is the $m\times1$ additive measurement noise vector. The specific equations for the measurement updates are presented below:

$$K_k = \frac{\bar{P}_k H^T}{(H\bar{P}_k H^T + R)}, \tag{15}$$

$$\hat{\boldsymbol{x}}_{\boldsymbol{k}} = \hat{\boldsymbol{x}}_{\boldsymbol{k}}^{-} + K_k(\boldsymbol{z_k} - H\hat{\boldsymbol{x}}_{\boldsymbol{k}}^{-}), \tag{16}$$

$$\hat{P}_k = (I - K_k H)\bar{P}_k, \tag{17}$$

$n\times m$ matrix K in equation 15 is chosen to be the gain that minimizes the posterior error covariance $\hat{P}_k$.

In this work, measurements z_k are the most likely x and y coordinates of the target object block in current compensated frame. Therefore, the observation matrix H is defined as [1 0 0] in each direction. The location of object blocks is measured by three-step-search (TSS) BM algorithm is used due to its simplicity and efficiency. Measurement noise v_k in equation 14 represents noise characteristics of the sensor. Since only the coordinate of the target object is measured from block matching, the covariance matrix, R, of v_k is represented by $[\sigma_z^2]$, where scalar value σ_z^2 is the determined empirically.

4. FUSION OF MOTION DETECTION ALGORITHMS FOR REGION RECOVERY

Given the results from temporal differencing and motion tracking, image regions are re-inspected to group moving regions from motion data. In this work, colour image segmentation algorithm in [15] is applied to group pixels in the image into coherent atomic regions at first. Then the degree of the motion activity, within each region is measured by the percentage of motion pixels. We use a 2-d vector $[u, v]$ to represent the percentage of motion pixels from temporal differencing and Kalman Filter based BM respectively. Since we have little information concerning the percentage of foreground pixels of the regions of change, except that they are expected to be significantly larger than zero, we do not attempt to analyse regions of changes, but consider the static regions instead. It is reasonable to assume that the motion activity measured by two algorithms in static regions can be modelled by a bivariate normal distribution $N(\mu_u, \mu_v, {\sigma_u}^2, {\sigma_v}^2, \rho)$. The parameters $\mu_u, \mu_v, {\sigma_u}^2, {\sigma_v}^2$ can be estimated using a robust k-medoids clustering algorithm [16]. Given $k = 2$, the motion activity data set $[u, v]$ are split into two clusters. Since in our case where the amount of change represents less than 10% of the image, the data set around the cluster with higher density can be considered as normally distributed and selected in parameter estimation. Therefore, the parameters estimation from the selected normal data becomes straightforward.

Once all the parameters $\mu_u, \mu_v, {\sigma_u}^2, {\sigma_v}^2$ are estimated, a two threshold values, T_{low}, T_{high} are computed to produce three classes of regions. A point that has a distance to the mean of the selected normal data, (μ_u, μ_v) smaller than T_{low} from the rest of the sample population of $2D$ points is said to have higher probability of being a static region due to lower motion activity detected with both algorithms. On the hand, point further away from the centroid means high motion activity in the segmented region detected with either algorithm or both algorithms. Since the variances in each direction of bivariate normal distribution are different, computing distances using standard Euclidean distance metric is not always beneficial. In such case, Mahalanobis distance should be used to classify a segmented region as moving region or static background. The Mahalanobis distance d_m of a bi-variate data-point $\boldsymbol{X} = (u, v)$ from the mean of the selected normal data, $\boldsymbol{\mu} = (\mu_u, \mu_v)$ is:

$$d_m(\boldsymbol{X} - \boldsymbol{\mu}) = \sqrt{(\boldsymbol{X} - \boldsymbol{\mu})^T S^{-1} (\boldsymbol{X} - \boldsymbol{\mu})} \tag{18}$$

where S is the covariance matrix.

Since Mahalanobis distance of the selected normal data follows a Chi-Square distribution with 2 degrees of freedom, T_{low}, T_{high} can be computed as in [17]. All regions above the T_{high} are retained as moving regions (class H), and all regions below T_{low} are rejected as static regions (class L). The remaining regions (class M) are retained only if they are adjacent to class H regions or are connected to class H regions via other class M regions. The advantage of applying hysteresis is that it incorporates spatial context into the region recovering decision, and effectively enables isolated (noisy) medium strength regions to be eliminated without fragmenting large regions containing low strength sections.

5. EXPERIMENT RESULTS

To demonstrate the robustness and validity of the proposed algorithm, we used the camera motion estimation method proposed in [18] to separate the camera motion from the object motion first. Then we compared the following methods: global temporal differencing method proposed by Rosin [4], local temporal differencing method proposed by Aach in [5], difference accumulation method proposed by Leng [6], and hybrid method of temporal differencing and optical flow method proposed by Tian [10]. Since there are no existing video datasets available specifically designed for video object segmentation under non-stationary camera with affine camera motions, 5 video sequences under camera translation and 4 video sequences under rotation in Figure 1 are used in our experiments. The image size for each video sequence is 896×504 pixels. The results of algorithms are compared to ground truth images which are obtained from manual segmentations done by human users. To evaluate the similarity between the segmentation and the ground truth, F_1 score [19], as a trade-off between precision and recall, is used.

$$F_1 = 2\frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}}, \tag{19}$$

$$precision = \frac{\text{TP}}{\text{TP} + \text{FP}}, recall = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{20}$$

where TP is the number of the true positives pixels which are correctly classified foreground pixels. FP is the number of background pixels, wrongly classified as foreground pixels. FN is the number of foreground pixels, wrongly classified as background pixels. 10 pairs of consecutive frame in each sample video are tested with those algorithms and F_1 score from each pair against a ground truth is calculated.

Experiment results measured with F_1 score from moving vehicles and pedestrian video sequences are shown in figure 2 and 3. It shows that proposed method achieved highest F_1 score in both test cases, close to 80%. Rosin's method has the worst performance overall due to its global thresholding. Aach's local method works slight better in comparison to Rosin's method. However, both methods have poor performance in detecting complete shapes of moving objects as shown in Figure 4 and 5. Also as shown in figure 4, both methods have poor performance in object localisation along the boundaries of pedestrian due to registration noise. Leng's method detects slow motion of pedestrian by accumulating past N frame differences but such accumulation also expands the actual changed region when the object is moving fast, such as in figure 5(d). Tian's method has better detection rate in terms of F measure but still cannot track the motion

accurately if objects stop, are occluded, or move fast as shown in figure 4(e) and figure 5(e).

6. CONCLUSION

The proposed algorithm separates the background interference and foreground information effectively with non-stationary camera and detect the local moving object accurately. It addresses the issues of uncertainty of motion, robustness to noise presence. The effectiveness of the proposed algorithm to robust detect salient motion is demonstrated for a variety of real environments.

(a) Pedestrian 1_T (b) Pedestrian 2_T (c) Traffic_T

(d) Carpark_T (e) Pedestrian 3_T (f) Pedestrian 4_R

(g) Traffic_R (h) Pedestrian 5_R (i) Hopspital_R

Figure 1 Testing Videos with camera translation in (a)-(e) and with camera rotation in (f)-(i)

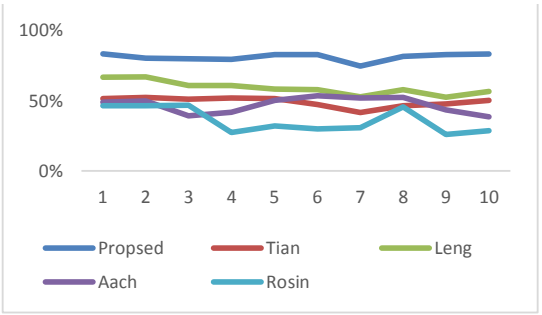

Figure 2 F1 score from Pedestrian 2_T

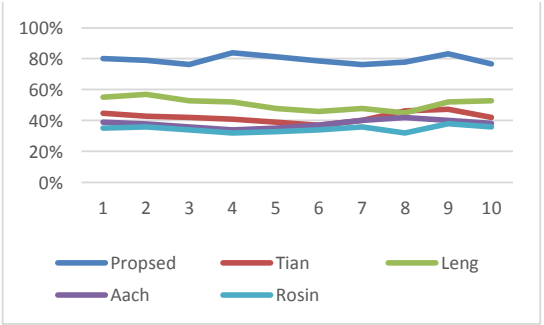

Figure 3 F1 score from Traffic_T

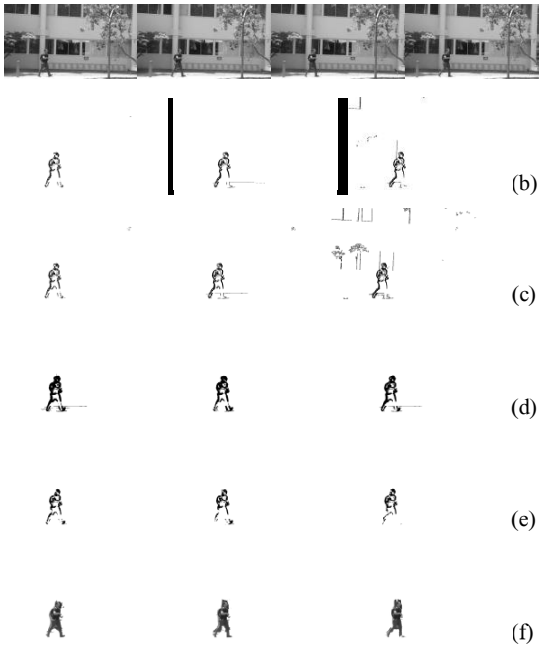

Figure 4: motion detection results comparison. (a) original frames 14-17 (b) Rosin's method (c) Aach's method (d) Leng's method (e) Tian's method (f) Proposed approach.

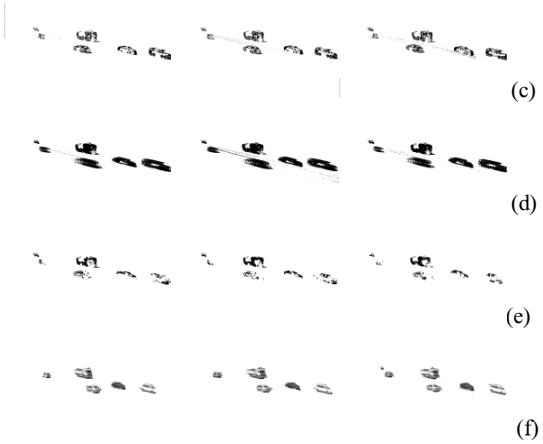

Figure 5: motion detection results comparison. (a) original frames 14-17 (b) Rosin's method (c) Aach's method (d) Leng's method (e) Tian's method (f) Proposed approach.

REFERENCES

[1] L. Wixson, "Detecting salient motion by accumulating directionally-consistent flow.," IEEE transactions on pattern analysis and machine intelligence, vol. 22, no. 8, pp. 774-780, 2000.

[2] Y. Zhen and Y. Chen, "A real-time motion detection algorithm for traffic monitoring systems based on consecutive temporal difference," in Asian Control Conference, 2009.

[3] N. S. Love and K. Chandrika., "An empirical study of block matching techniques for the detection of moving objects," in Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, Livermore, 2006.

[4] P. Rosin, “Thresholding for change detection,” in Sixth International Conference on. IEEE omputer Vision, 1998.

[5] T. Aach,, A. Kaup and R. Mester, “Statistical model-based change detection in moving video., 31(2), pp.165-180.,” Signal processing, 1993.

[6] L. Bing and Q. Dai, “Video object segmentation based on accumulative frame difference,” in Picture Coding Symposium , 2007.

[7] M. Narayana, A. Hanson and E. Learne, “Coherent motion segmentation in moving camera videos using optical flow orientations,” in The IEEE International Conference on Computer Vision, 2013.

[8] J. Hariyono, V.-D. Hoang and K.-H. Jo, “Moving object localization using optical flow for pedestrian detection from a moving vehicle,” The Scientific World Journal , vol. 10, no. July, 2014.

[9] J. Kim, X. Wang, H. Wang and C. Zhu, “Fast moving object detection with non-stationary background,” Multimedia tools and applications, vol. 67, no. 1, pp. 311-335, 2013.

[10] Y. Tian and A. Hampapur , “Robust salient motion detection with complex background for real-time video surveillance,” in Seventh IEEE Workshops In Application of Computer Vision, 2005.

[11] J. Philip, B. Samuvel, K. Pradeesh and N. Nimmi, “A comparative study of block matching and optical flow motion estimation algorithms,” in Annual InternationalIn Emerging Research Areas: Magnetics, Machines and Drives (AICERA/iCMMD), 2014.

[12] P. L. Rosin and T. J. Ellis, “Image difference threshold strategies and shadow detection,” in BMVC, 1995, July.

[13] T. Aach,, A. Kaup and R. Mester, “Statistical model-based change detection in moving video., 31(2), pp.165-180.,” Signal processing, 1993.

[14] . X. R. Li and V. Jilkov, “Survey of maneuvering target tracking: dynamic models,” in AeroSense 2000 International Society for Optics and Photonics, 2000.

[15] X. Le and R. Gonzalez, “A Consistent, Real-Time Image Segmentation for Object Tracking,” in IEEE International Conference In Digital Image Computing: Techniques and Applications (DICTA), 2016.

[16] H. S. Park and C. H. Jun, “A simple and fast algorithm for K-medoids clustering,” Expert Systems with Applications, vol. 36, no. 2, p. 3336–3341, 2009.

[17] P. J. Rousseeuw, and B. C. Van Zomeren, “Unmasking multivariate outliers and leverage points,” Journal of the American Statistical association, vol. 85, no. 411, pp. 633-639., 1990.

[18] X. Le and R. Gonzalez, “A robust region-based global camera estimation method for video sequences.,” in Signal Processing and Communication Systems (ICSPCS), 2013 7th International Conference on., 2013.

[19] D. M. Powers, “Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation.,” Journal of Machine Learning Technologies. , vol. 2 , no. 1, p. 37–63, 2001.