An Algorithm for Detecting the Exact Regions of Moving Objects in Video Frames

Salar Fattahi , Masoumeh Azghani , Farokh Marvasti
Dept. of Electrical Engineering, Sharif University of Technology, Iran
salarfatahi88@gmail.com , elmaazghani@yahoo.com , fmarvasti@gmail.com

***Abstract* --- Motion detection is of paramount importance in video surveillance systems. In this paper, a novel algorithm is proposed to extract the exact boundaries of moving objects in video frames. Using the concepts of Cross-Correlation and Edge Detection, we combine two well-known motion detection methods to extract the moving regions more accurately. Also, we modify these two methods in terms of accuracy and processing time in order to make them suitable for algorithm. The experimental results prove that the proposed method significantly outperforms the other state-of-the-art schemes in terms of both objective and subjective performance. Another advantage of the suggested method over the others is its relatively low computational complexity and high implementation speed.**

Keywords—Motion Detection, Background Subtraction, Moving Object Extraction, Video Processing

I. Introduction

Video surveillance systems have been used in a wide range of science and technology applications in computer vision, including human activity recognition, intelligent video surveillance, and endangered species conservation [1].

The motion detection algorithms can be grouped as optical flow [2], frame difference [3] and background subtraction [4], [12]. The optical flow methods are complex and are extremely sensitive to noise. The frame difference methods have less computational complexity but do a poor job of extracting the complete regions of the moving objects. The background subtraction methods have less computational complexity than optical flow methods and have better results than the frame difference algorithms. Hence, the proposed algorithm in this paper is based on the background subtraction method. The Running Average (RA) is a method based on background subtraction. In RA, the adaptive background model is attained using previous background frame and incoming video frame [4]. In this method, some artificial trails are created behind moving objects in the background model, thus it is not applicable in dynamic backgrounds. The Gaussian Mixture Models (GMM) [5] simulates the background by samples which are weighted and mixed according to Gaussian distributions to adapt to the dynamic changes of the background. The disadvantage of the GMM method is its low convergence speed especially for the crowded scenes. The Kernel Density Modeling (KDM) is a typical nonparametric background modeling method [6] which does not assume the Gaussian model for pixel distribution and directly estimates the background probability density function. However, KDM method does not have precise outcome. Zivkovic [7] proposed an improvement to reduce the computational time in KDM and GMM methods. He used Bayesian formula to determine the sufficient number of Gaussian modes for each pixel. Gorur [8] provided more computation time reduction for GMM method by an orthogonal approach that minimizes floating point computation. He proposed to update the weights of Gaussian modes only once in T_w frames. Although both of these methods reduced computation time significantly, they failed to enhance the accuracy of the outcomes. In order to increase the accuracy of moving object detection, McHugh [9] proposed an iterative algorithm for modeling the foreground in addition to the background. In each iteration step, a small neighborhood around each pixel is exploited to construct the foreground model. This method improves the accuracy of the resulted outcomes at the expense of more computational complexity. Zhao [10] has proposed a novel algorithm for discriminating the candidate left objects from moving ones. It uses the contour correlations to find the static objects. Then, it eliminates ghosts in the candidate left objects by comparing the correlation of their inner and outer contours. Although this method is very useful for extracting the left objects, it fails to truly extract moving objects. Hati [11] proposed an Intensity Range (IR) based algorithm that finds lower and upper bounds for background pixels by using the absolute differences of pixels in the initial frames within a window size. The drawback of this method is that the initial frames must be free of moving objects which may not be available most of the time.

In this paper, a combination of the modified KDM and RA methods called MKDRA is proposed for efficient motion detection. In proposed method, an edge detection algorithm together with a morphological operation is exploited to extract the exact regions of the moving objects, besides reducing the noise.

978-1-4799-5359-2/14/$31.00

The simulation results confirm the outperformance of the KDMRA method against its other counterparts based on both qualitative and quantitative measurements.

The remainder of this paper is organized as follows. Section II describes the KDM and RA methods. The proposed MKDRA method is illustrated in section III. Section IV gives experimental results. The conclusions are given in Section V.

II. KDM AND RA METHODS

In this section, the Kernel Density Modeling and the Running Average methods are described.

A. Kernel Density Modeling (KDM)

KDM is a nonparametric statistical modeling method which estimates the Probability Density Function (PDF) directly from the data without any further assumption about the corresponding distribution.

The underlying PDF is estimated as:

$$f(x) = \sum_i \alpha i K(x - xi) \tag{1}$$

where K is "Kernel Function" which can typically be a Gaussian function centered at data points, x_i , and α_i 's are the weighting coefficients of the Kernel Functions.

Given pixels $\{x_i\}_{i=1,\ldots N}$ from a distribution with density function $f(x)$, an estimation of the background probability density function can be calculated as:

$$f'(x) = \frac{1}{N} \sum_{i=1}^{N} K_\sigma (xt - xi) \tag{2}$$

$$\sigma = \frac{m}{0.68\sqrt{2}} \tag{3}$$

where K_σ is a kernel function with a bandwidth σ and m is the median of $|x_i - x_{i+1}|$ for i=1…N-1.

The pixel belongs to the foreground if $f'(x) < th$, where th is a global threshold for all of the frames.

The PDF is calculated based on the N most recent samples. Therefore, the adaptation of the model can be achieved by adding new samples and ignoring the older ones. Another approach for classifying the pixels is suggested in [5] which calculates the Mahalanobis Distance to find the pixels that are matched to the background PDF function. The Mahalanobis Distance of a multivariate vector $x = (x_1, x_2..., x_N)^T$ from a set of data with mean $\mu = (\mu_1, \mu_2, ..., \mu_N)^T$ and covariance matrix S is defined as:

$$D_M(x) = \sqrt{(x - \mu)^T S^{-1} (x - \mu)} \tag{4}$$

In this paper, we use Mahalanobis Distance to classify the pixels.

B. Running Average (RA)

In the Running Average method, the previous background frame $B_{t-1}(x,y)$ and the new incoming frame $I_t(x,y)$ are integrated to achieve the current background image. The adaptive background model is attained using a simple adaptive filter as follows:

$$B_t(x, y) = (1 - \beta)B_{t-1}(x, y) + \beta I_t(x, y) \tag{5}$$

where β is an empirically adjustable parameter. While a larger β leads to a faster background updating, smaller β causes artificial trails behind moving objects in the background model.

The background frame is subtracted from the current one and thresholded to obtain the moving regions. The binary motion detection mask $D(x,y)$ is defined as follows:

$$D(x,y) = \begin{cases} 1, & if\ |It(x,y) - Bt(x,y)| > \tau \\ 0, & if\ |It(x,y) - Bt(x,y)| < \tau \end{cases} \tag{6}$$

where $I_t(x,y)$ is current video frame, $B_t(x,y)$ is the current background model, and τ is an experimental threshold.

III. THE PROPOSED KDMRA METHOD

In this section, KDMRA method is presented. The block diagram of the KDMRA method is depicted in Figure 1.

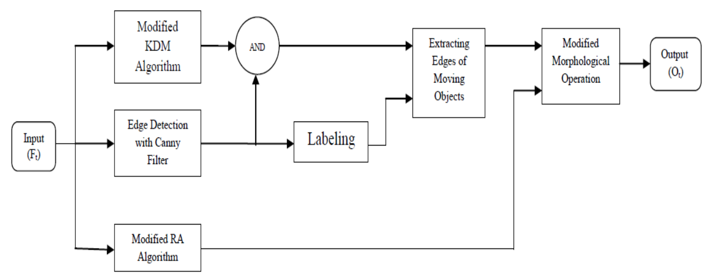


Figure1. The Block Diagram of the Proposed Method

According to the block diagram, we can define this algorithm in 6 steps:

1) The modified KDM algorithm is applied to the incoming frame F_t to obtain Primary binary Motion Detection mask (MKD_t)

2) The Canny edge detector [13] is applied to the F_t to extract the edges of the frame (E_t). Note that we use Canny filter in this algorithm because it is more robust to noise and more likely to detect true weak edges.

3) The pixel wise AND operand is used to combine MKD_t with E_t to get the edges of moving objects named ME_t.

4) The edges of the frame E_t are labeled as LE_t and combined with ME_t. To extract the true and complete edges of the moving objects, the locations of nonzero elements in ME_t are determined, firstly. Then, in LE_t, the pixels that have the same labels with the ones in these locations are kept and the rest of them are set to zero. The output of this step is called CLE_t.

Up to this stage, we have extracted the complete edges of moving objects.

There is still an important problem in this algorithm that can cause a big error in CLE_t. Suppose that the primary motion detection algorithm erroneously detects a small region as a moving part in MKD_t. This region can cover some edges of the still objects which would be extended in CLE_t. To eliminate these unwanted edges from CLE_t, we use next two steps:

5) Modified RA method is applied to F_t (MRA_t).

6) MRA_t is combined with CLE_t using the modified morphological operations.

To present the proposed morphological operation, a quantity called *motion ratio* is defined as:

$$\frac{\sum_{c=i}^{j} MRA_t(r,c)}{j-i+1} \tag{7}$$

where i and j are the column indices of the two pixels with the same labels in the r^{th} row of the matrix CLE_t. The operation extracts the whole area of the moving objects from their incomplete regions obtained from the MRA approach. The pixels between the two equally labeled pixels in each row (i and j) are considered as moving ones only if the motion ratio of the pixels between these two is above a predefined threshold, th_{morph}.

Now consider the problem that we mentioned in step 4. Suppose some edges are extracted wrongly in CLE_t. These edges are not the parts of moving objects, so the motion ratio around these edges is small and they will be eliminated automatically in this step. The output of this stage (O_t) contains the exact regions of moving objects. Simulation results show that our method significantly reduce False Positive pixels as well as increasing True Positive ones.

Next, we present our modified KDM and RA methods.

A. Modified Kernel Density Modeling

The conventional KDM is time consuming due to its high computational complexity. To improve the speed of the algorithm, a modification of the KDM method is introduced here. This method would be applied in the first step to estimate the approximate regions of the moving objects. Hence, instead of extracting the moving pixels, we try to estimate the moving blocks. To achieve this goal, the *Cross-Correlation* is defined as:

$$C.C(X,Y)=\frac{X.Y}{|X|.|Y|} \tag{8}$$

where ". " represents the inner vector product.

At First, each incoming frame is divided into 8*8 blocks. Then, the values of $C.C(F_n(i,j),F_{n-1}(i,j))$ are computed , where $F_n(i,j)$ and $F_{n-1}(i,j)$ are the $(i,j)^{th}$ blocks of F_n and F_{n-1}, respectively. The Cross-Correlation is a measure for the amount of similarities between corresponding blocks. If $C.C(F_n(i,j),F_{n-1}(i,j))$ is small, the two blocks are considered to be less similar to each other and vice versa. Note that we store Cross-Correlations of frames F_n and F_{n-1} in a matrix *CCM* which will be updated and used in the modified RA method.

By thresholding the Cross-Correlation matrix, we classify the blocks of the current frame as Suspected Blocks (SB) or Dynamic Blocks (DB). The DBs are straightforwardly considered as moving blocks, while the moving blocks among the suspected ones are determined using the KDM algorithm. Hence, the computational complexity is reduced by applying the KDM only to the SBs. Note that we need to store the last *N* samples, $\{x_i\}_{i=1,\dots,N}$, for every pixel in the frame, either DBs or SBs, since some DBs may become static in the future.

It is important to apply KDM to SBs since the behavior of these blocks should be observed during a long period.

As mentioned earlier, Mahalonobis distance method is exploited in this paper for the task of pixel classification. If the sizes of the SBs are small enough compared to the moving objects, we can consider each static block as a super pixel with the average of the pixels as its graylevel value. The Mahalanobis based KDM method is then applied to the super pixels, reducing the computational complexity.

B. Modified Running Average

The RA method is not accurate enough in extracting the exact regions of moving objects. This is because the parameter β in (5) is a fixed value for all pixels in all frames. β should be an adaptive value depending on the motion speed. For slow motion regions, the background should be updated slowly and as a result, β should be small while for fast motion blocks, β should be large to update the background fast.

To that goal, we use Cross-Correlation values to assign an independent β for each block of the frame according to environmental movement of that block.

If the value of Cross-Correlation between the $(i,j)^{th}$ blocks in F_n and F_{n-1} is small, the background should be updated faster for this block and β should be a large value and vice versa. To match the values of $C.C$s to appropriate β, we can use a linear mapping that maps $C.C$s to values of β in the range $[\alpha_1, \alpha_2]$. β is set to α_2 for minimum $C.C$ (maximum motion) blocks and it is set to α_1 for maximum $C.C$ (minimum motion) blocks. In our simulations, we realized that the best values for α_1 and α_2 are 0.1 and 0.9, respectively.

IV. Experimental Results

In this section, the experimental results are presented. Figure 2 shows the improvement of RA method by selecting the updating parameter β adaptively. By using this approach, the shadows of the moving objects are eliminated and the extracted objects are more accurate.

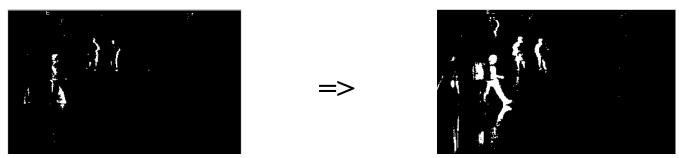
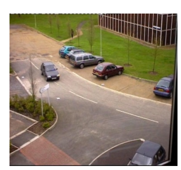
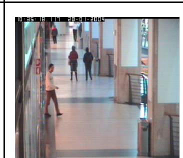
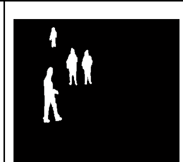


Figure 2. left: RA method, right: the modified RA

The proposed MKDRA method and several other state-of-the-art methods are simulated for three video sequences, *PETS 2000*, *PETS 2001* and *EnterExitCrossingPaths*. The simulation results are given in Figure 3.

	Pets2000-0130-0150	EnterExitCrossing Paths1cor	pets2001-dataset1-0510-0540
frame			
Ground Truth			

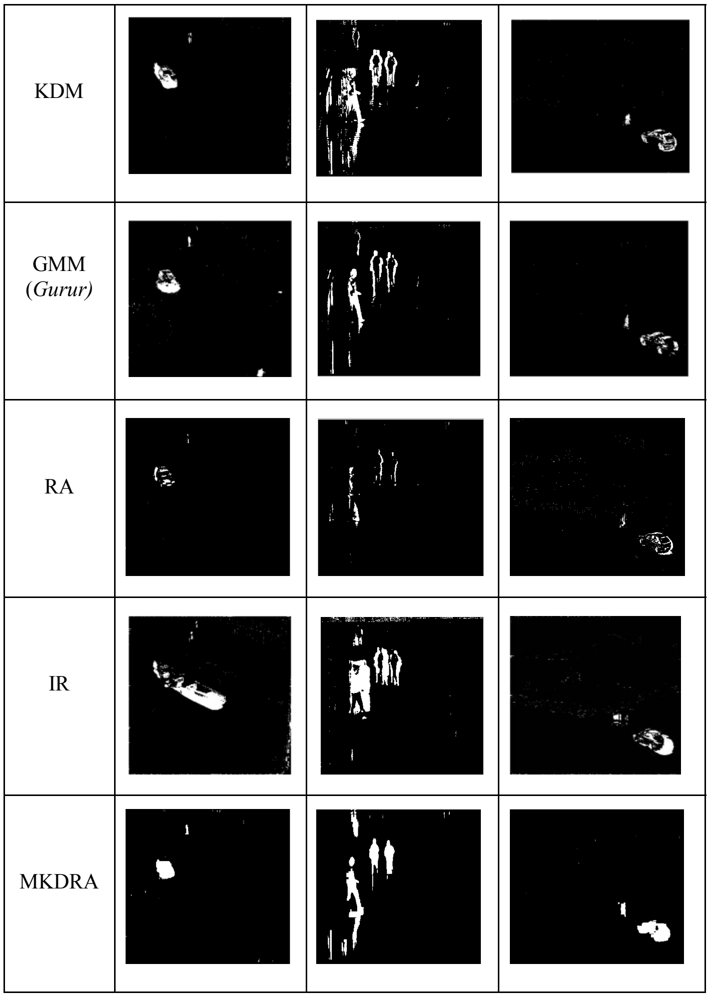


Figure 3. Extracted objects using proposed method and other methods, KDM: Kernel Density Method [6], GMM *(Gurur)*: Gaussian Mixture Model (Gurur's Method) [8], RA: Running Average [4], IR: Intensity Range [11], MKDRA: Modified Kernel Density and Running Average (proposed method)

As one can see in Figure 3, the proposed method yields satisfactory results for three video sequences. Note that in none of the simulations, we use separate initial free-of-moving object background frames; a hypothesis that is applicable to most practical cases. Therefore, in our simulations, the IR method has not obtained acceptable results.

In "pets2000-0130-0150", the MKDRA method has the better result than the others and extracts the complete moving regions. Also, the noise has been reduced in the proposed method.

The suggested method is also capable of extracting the slow moving objects. This fact can be perceived in the "EnterExitCrossingPaths1cor" sequence which contains low speed movements.

In "pets2001-dataset1-0510-0540", the true extraction of moving objects is more challenging because of noise and slow motion. The MKDRA method has produced better results compared to the other ones.

In order to have an objective comparison, some quantitative metrics are used which are defined as follows:

$$Recall = tp / (tp + fn) \tag{9}$$

$$Precision = tp / (tp + fp) \tag{10}$$

$$F1 = 2(Recall)(Precision) / (Recall + Precision) \tag{11}$$

$$Similarity = tp / (tp + fp + fn) \tag{12}$$

where tp, fp, and fn are defined as:
(tp) True Positive: Correctly Identified Pixels
(fp) False Positive: Incorrectly Identified Pixels
(fn) False Negative: Incorrectly Rejected Pixels

Table 1.Quantitative Evaluation

Sequence	Evaluation	KDM	GMM *(Gurur)*	RA	IR	MKDRA
PETS 2000-0130-0150	Similarity	0.4728	0.4214	0.1849	0.0489	0.7590
	F1	0.6420	0.5930	0.3122	0.0933	0.8630
	Precision	0.6808	0.5737	0.6876	0.0623	0.9084
	Recall	0.6075	0.6135	0.2019	0.1857	0.8219
Enter Exit Crossing Paths	Similarity	0.3318	0.3458	0.0011	0.1794	0.5100
	F1	0.4982	0.5139	0.0022	0.3042	0.6758
	Precision	0.4856	0.5382	0.0725	0.2508	0.6200
	Recall	0.5116	0.4916	0.0011	0.3867	0.7421
Pets2001-dataset1-0510-0540	Similarity	0.3266	0.3327	0.0767	0.3326	0.6821
	F1	0.4924	0.5298	0.1424	0.3923	0.8110
	Precision	0.7815	0.8212	0.7534	0.3600	0.8034
	Recall	0.3595	0.3510	0.0787	0.2195	0.8188

The quantitative measurements are given in Table 1. According to our simulations, the proposed MKDRA method outperforms the other state-of-the-art methods in the above-mentioned quantitative metrics. The main reason is that the MKDRA has reduced False Positive pixels due to the effective reduction of noise. Also, this approach extracts the exact region of moving objects and fills holes within the detected ones. As a result, the True Positive pixels are increased, while the False Negative ones are decreased.

As a trustable measure of complexity, the simulation time of various methods are compared in Table 2.

Table 2.Processing Time Evaluation

Sequence	KDM	GMM *(Gurur)*	RA	IR	MKDRA	KDM (In our method)	RA (In our method)
PETS 2000-0130-0150	16.7 f/s	17.0 f/s	69.2 f/s	12.6 f/s	17.8 f/s	56.5 f/s	68.1 f/s
EnterExitCrossing Paths	82.4 f/s	73.1 f/s	225.8 f/s	56.5 f/s	80.1 f/s	240.0 f/s	221.4 f/s
Pets2001-dataset1-0510-0540	17.4 f/s	19.4 f/s	80.2 f/s	13.1 f/s	20.5 f/s	60.1 f/s	75.3 f/s

The processing rate of the MKDRA method is better than those of the KDM, GMM, and IR methods in non-crowded sequences. However, for *EnterExitCrossingPaths* as an example of a crowded sequence, the processing rate of the KDM outperforms that of MKDRA. In the last two columns, the processing speeds of the modified KDM and RA methods are presented. The modified KDM method suggested here is 3 times faster than the original KDM. This is because we extract the moving blocks rather than the moving pixels.

V. Conclusion

This paper presents an improved moving object detection algorithm based on modified KDM and RA methods. This algorithm not only extracts the regions of the moving objects accurately using the edge detection, but also it relatively improves the processing rate. Furthermore, it divides images into small blocks and use Cross-Correlation to eliminate the unnecessary set of calculations in KDM as well as improving the RA method by using adaptive updating parameter β. The simulation results demonstrate that the proposed MKDRA method attains the most pleasing outcomes from both quantitative and qualitative points of view. However, the overall speed of the method is not sufficiently high in crowded scenes. In the future, we will focus on improving the computational efficiency of the proposed method.

REFERENCES

[1] W. Hu, T. Tan, L. Wang, and S. Maybank, "A survey on visual surveillance of object motion and behaviors" IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, 34(3), 334-352. (2004)

[2] P. Gao, X. Sun, and W. Wang, "Moving object detection based on kirsch operator combined with Optical Flow", 2010 International Conference on Image Analysis and Signal Processing (IASP), IEEE. (2010, April)

[3] J. R. Bergen, P. J. Burt, R. Hingorani, and S. Peleg, "A three-frame algorithm for estimating two-component image motion" IEEE Transactions on Pattern Analysis and Machine Intelligence, 14(9), 886-896. (1992)

[4] C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland, "Pfinder: Real-time tracking of the human body" IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(7), 780-785, (1997)

[5] N. Friedman, and S. Russell, "Image segmentation in video sequences: A probabilistic approach" Proceedings of the Thirteenth conference on Uncertainty in artificial intelligence (pp. 175-181). Morgan Kaufmann Publishers Inc., (1997, August)

[6] A. Elgammal, D. Harwood, and L. Davis, "Non-parametric model for background subtraction" In Computer Vision—ECCV (pp. 751-767), Springer Berlin Heidelberg, (2000)

[7] Z. Zivkovic, and F. van der Heijden, "Efficient adaptive density estimation per image pixel for the task of background subtraction" Pattern recognition letters, 27(7), 773-780. (2006)

[8] P. Gorur, and B. Amrutur, "Speeded up gaussian mixture model algorithm for background subtraction" 2011 8th IEEE International Conference on In Advanced Video and Signal-Based Surveillance (AVSS), IEEE (2011, August)

[9] J. M. McHugh, J. Konrad, V. Saligrama, and P. Jodoin, "Foreground-adaptive background subtraction" Signal Processing Letters, IEEE, 16(5), 390-393, (2009)

[10] H. Zhao, H. Yang, and S. Zheng, "An Efficient Method for Detecting Ghosts and Left Objects in Intelligent Video Surveillance" 2nd International Congress on Image and Signal Processing, 2009, CISP'09, IEEE

[11] K. K. Hati, P. K. Sa, and B. Majhi, "Intensity Range Based Background Subtraction for Effective Object Detection" Signal Processing Letters, IEEE,20(8), 759-762, (2013)

[12] L. Maddalena, and A. Petrosino, "A self-organizing approach to background subtraction for visual surveillance applications" IEEE Transactions on Image Processing, 17(7), 1168-1177

[13] M. Sonka, V. Hlavac, and R. Boyle, "Image processing, analysis, and machine vision" (Vol. 3). Toronto: Thomson, (2008)