

Automatic foreground detection based on KDE and binary classification

Mohammed Lahraichi, Khalid Housni, Samir Mbarki

MISC Laboratory, Ibn Tofail University, Faculty of Sciences Kenitra, Morocco

Article Info

Article history:

Received Nov 21, 2018

Revised Jan 21, 2019

Accepted Feb 28, 2019

Keywords:

Energy function

Graph cut

Moving object detection kernel density estimation

ABSTRACT

In the recent decades, several methods have been developed to extract moving objects in the presence of dynamic background. However, most of them use a global threshold, and ignore the correlation between neighboring pixels. To address these issues, this paper presents a new approach to generate a probability image based on Kernel Density Estimation (KDE) method, and then apply the Maximum A Posteriori in the Markov Random Field (MAP-MRF) based on probability image, so as to generate an energy function, this function will be minimized by the binary graph cut algorithm to detect the moving pixels instead of applying a thresholding step. The proposed method was tested on various video sequences, and the obtained results showed its effectiveness in presence of a dynamic scene, compared to other background subtraction models.

Copyright © 2019 Institute of Advanced Engineering and Science.
All rights reserved.

Corresponding Author:

Mohammed Lahraichi,
MISC Laboratory,
Faculty of Sciences,
Ibn Tofail University, Kenitra, Morocc.
Email: lahraichi.mohamed@gmail.com

1. INTRODUCTION

The separation of moving pixels from their background is an essential phase in many computer vision fields, especially, in video surveillance, traffic monitoring, activity recognition, etc. The main idea is to create a statistical model of the background, aiming to generate a representation of a background image based on previous frames by using density functions [1], either on each pixel or by regions. Then, this representation is compared against the input frame to get a binary mask image which represents the position of the moving objects. Yet, the background is not always static in general, so the model must be robust and more adaptive for the purpose to overcome some frequent issues successfully, such as gradual or sudden illumination change, non stationary background [2].

Different background modeling techniques have been proposed to address the previous limitations [2]. The traditional ones are based on pixel intensity, which exploit only the intensity value to decide if a pixel belongs to the background or the moving objects. Despite their promising performances, they generate some misclassified pixels, especially if the background and the foreground have the same color, and because they also ignore the spatial dependencies of neighboring pixels. While, models based on texture features [3] have demonstrated a certain degree of success in exploiting the spatial correlation, they consider discriminative texture measure as features to distinguish moving pixels from the background. Although, they still have some shortcomings like the use of a threshold to detect the moving pixels. Recently, several methods based on deep learning have appeared [4]. Which aim at handling all above limitations, However, they require a training phase with several annotated examples, that needs more computational time.

In order to tackle some of these issues, firstly, we generate a probability image using KDE method, and then, instead of using a threshold to segment this image into foreground and background, this binary

segmentation of the probability image is performed by minimizing an energy function using the graph cut algorithm, which exploits the spatial correlation of neighboring pixels.

The remainder of this paper is organized as follows: The second section presents theoretical reviews, the third section shows research method, in which we present how to construct probability image using KDE, and describe our moving object detection method using a graph cut algorithm; the fourth section discusses the experiments results and finally the conclusion.

2. THEORETICAL REVIEW

In order to detect moving objects using intensity pixel value, Stauffer et al. [5] have proposed to model each pixel with a mixture of K Gaussians distributions. This model is used to determine whether a pixel belongs to the background or the foreground by comparing it with each Gaussian distribution, the initialization of the Gaussians distribution is made by using an Expectation Maximization algorithm, however, the number of Gaussians distributions and the parameters of each distribution must be initialized. Zivkovic in [6] improves upon the widely used parametric Gaussian Mixture Model system by introducing an on-line clustering algorithm to separate foreground clusters from background ones. In order to exploit the spatial information and deal with highly dynamic scenes, Haiying et. al. [7] have proposed a modified Gaussian mixture background model based on the spatial-temporal distribution which uses time and space distribution information. While [8] proposes an effective scheme for modelling and updating a background adaptively in dynamic scenes focus on statistical learning, and [9], proposes a new method as an adaptation of the MOG approach for detecting the moving objects, where the foreground is extracted by considering the HSV color space.

To overcome the parameters initialization of GMM, a non-parametric approach called Kernel Density Estimation (KDE) that can effectively adapt to a dynamic background was proposed [10]. However, KDE has to keep N frames in memory which is time consuming when N is large, Park et al. [11]. Used the Bayesian rule with the KDE method and applied histogram approximation to decrease the computational cost. Whereas, [12] follows a nonparametric background modeling paradigm, in which each location in a dynamic scene collects a set of samples on different spatial scales which occurred in the past time and in the neighborhood.

On the other hand, Kim et al. [13] have proposed a codebook method which uses a clustering technique to model the background and initializes codewords of codebooks to store background states, where codewords are a series of key color values. Next, Shah et al. [14] designed a Self-Adaptive CodeBook (SACB) background model. This model is a block-based structure using the close proximity of local neighborhoods. In this method, the exponential smoothing filter is adapted to keep the mean and variance values in order to automatically estimate the brightness boundary threshold and discolorations for each code word. Whereas [15], presents a novel dynamic codebook method to address such challenges. The dynamic codebook aims to significantly improve the conventional well-known codebook technique by introducing a technique to make a dynamic boundary of each codeword

In order to robustly deal with varying illumination conditions, the use of textures have been proposed on a block-wise processing approach and extended to the pixel level by Heikkilä [16], these methods use discriminative texture features to capture background statistics. These features are computed by using a Local Binary Pattern (LBP) in order to consider dynamic textures, the use of a Volume Local Binary Pattern (VLBP) operator is proposed in [17], which consists of concatenated LBP histograms from three orthogonal planes. So as to combine spatial and temporal informations, [18] have proposed a Spatio-Temporal Local Binary Pattern (STLBP) operator, consisting in a weighted sum of two consecutive LBP histograms, to alleviate the computational cost imposed by VLBP, LBP histograms provides a robust manner to cope with illumination changes in dynamic scenes. Nevertheless, they do not provide a principled manner to evaluate the distance of new observations to the background models. [19] Have presented a modified Local Binary Similarity Pattern (LBSP) descriptor to set up the background model in feature space, it calculated the LBSP descriptor by absolute difference which is different from LBP.

Deep learning, especially convolutional neural networks (CNNs) have recently been very popular and have been used successfully in moving object detection. The reference [20] proposes two robust encoder-decoder type neural networks that generate multi-scale feature encodings in different ways and can be trained end-to-end using only a few training samples. Where Wang and al. [21] have proposed a highly accurate semi-automatic method for segmenting foreground moving objects pictured in surveillance videos. They implement a end-to-end model based on a multi-resolution convolutional neural network (CNN) with a cascaded architecture. This one does not need a large number of examples to accurately fit the data. With the aim of addressing the complex nature of the dynamic scene in real surveillance task, the authors [22] have presented a simple and efficient vector-based method is proposed to address real surveillance challenges,

where the concept of linear dependence of vectors is used to build background models corresponding to each pixel, where linear independence is used to detect moving object from incoming video sequel. Babae et al. [23], propose a new approach to estimate background model from video sequences, where feature engineering and parameter tuning become unnecessary since the network parameters can be learned from data by training a single CNN that can handle various video scenes. For the training of the CNN, they employed randomly 5% video frames and their ground truth segmentations taken from the Change Detection challenge 2014 (CDnet 2014), and also used spatial-median filtering as the post-processing of the network outputs.

3. RESEARCH METHOD

a) Probability Image using Improved KDE

Kernel density estimation [10] is a non-parametric approach which estimates the probability density of a pixel using a sample of data. The calculation of the probability image using KDE is performed through a three steps, which will be explained in the following subsections.

3.1.1. Probability Density of a Pixel using KDE Model

Considering $\{x_1, x_2, \dots, x_n\}$ a sample of a pixel values from previous frames, in general, the estimation of probability density for pixel value x_t at time t , is given as:

$$P(x_t) = \frac{1}{n} \sum_{i=1}^n k_{\sigma}(x_t - x_i) \tag{1}$$

Where n denotes the number of samples σ represents the kernel function bandwidth, and $k(x)$ is a kernel function which should satisfy these three conditions:

- 1) $K(x) \geq 0$
- 2) $\int K(x) dx = 1$
- 3) $\int xK(x) dx = 0$ *should be symmetric*

There are several kernel functions, which satisfy the above conditions, such as Gaussian kernel, Epanechnikov, triangular and uniform kernel, in this paper; we choose the Gaussian one as kernel function, which can be described as:

$$K(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{x^2}{2\sigma^2}\right) \tag{3}$$

So using (3), the (1) becomes:

$$P(x_t) = \frac{1}{n\sqrt{2\pi\sigma^2}} \sum_{i=1}^n \exp\left(-\frac{(x_t - x_i)^2}{2\sigma^2}\right) \tag{4}$$

Where σ is estimated using the median or the average of $|x_i - x_{i+1}|$ for each consecutive pair (x_i, x_{i+1}) in the sample of pixel values [10]:

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

$$m = \frac{\sum_{i=1}^{n-1} |x_i - x_{i+1}|}{n} \quad \text{or} \quad m = \text{median}(|x_i - x_{i+1}|) \tag{5}$$

Each Gaussian kernel describes just one sample data, and σ is the same for all kernels of the same pixel.

3.1.2. Updating Background Model

In reality, the background is never completely static, but changes over time. So, it is necessary to adapt the background model to those changes, by updating the sample of pixel values in FIFO (First In First Out) order as follows:

Let x_t denotes the pixel value in the sample, and x_t is the current pixel value.

If x_t is the value of background pixel then: $x_t = \alpha x_t + (1 - \alpha) x_t$

Where α is an empirical weight often chosen as a tradeoff between stability and quick update [10]

In general, probability density reflects the variation of the pixel value, so, if this latter changes frequently during a period of time, that means its probability value will be small, and there is more possibility to be moving object pixel. By contrast, if the pixel keeps the same value, or changes a little, then it has a large chance to be a background pixel.

3.1.3. Probability Image

After calculating the probability density $p(x_t)$ using (1), each pixel value in probability image is computed as follows: $ctse \times P(x_t)$ ($ctse = 255$). The value of $ctse$ is set to 255, in order to transform probability value into a grayscale value.

Figure 1(b) shows an example of probability image, where the dark zone represents pixels with lower probability (moving object), and the white zone represents pixel with higher probability (background pixel), so the problem consists of separating the moving object from the background. KDE model uses a global threshold to segment the moving object from the background, and ignore the correlation that exists between the intensity of neighboring pixels. To address these limits, we exploit the performance of the graph cut algorithm to extract moving pixels from its background.

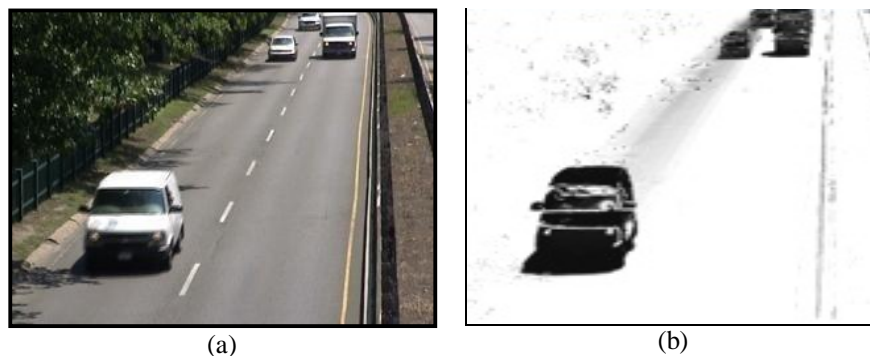


Figure 1. (a) Original image, (b): Estimated probability image

3.2. Moving Object Detection based on Graph Cut

The moving object detection problem can be viewed as a binary labeling task, label (1) for object pixels and (0) for background pixels or vice versa. The use of threshold to detects moving objects, is considered as a difficult task as the Figure 2 demonstrates, where the use of a lower threshold in (b) generate more false positive pixels, and the use of a high threshold eliminates true positive pixels, so to overcome this issue, we segment the pixels into background/moving pixel, through the minimization of an energy function using graph-cut.

3.2.1. Energy Function

Considered $X = \{x_1, x_2, \dots, x_N\}$ an observation set of a current probability image I in a video sequence, our objective is to assign a label from the binary set $Y = \{y_i / y_i \in \{0, 1\}\}$ to each pixel in I , which is equivalent to maximize the posterior probability $P(Y / X)$.

The Bayes law allows us to write the posterior probability $P(Y / X)$ as follows:

$$P(Y / X) = \frac{P(X / Y)P(Y)}{P(X)} \tag{6}$$

Since $P(X)$ is a constant term, so maximizing $P(Y / X)$ is equivalent to maximizing $P(X / Y)P(Y)$:

$$argmax_{y_p \in Y} (P(Y / X)) = argmax_{y_p \in Y} (P(X / Y)P(Y)) \tag{7}$$

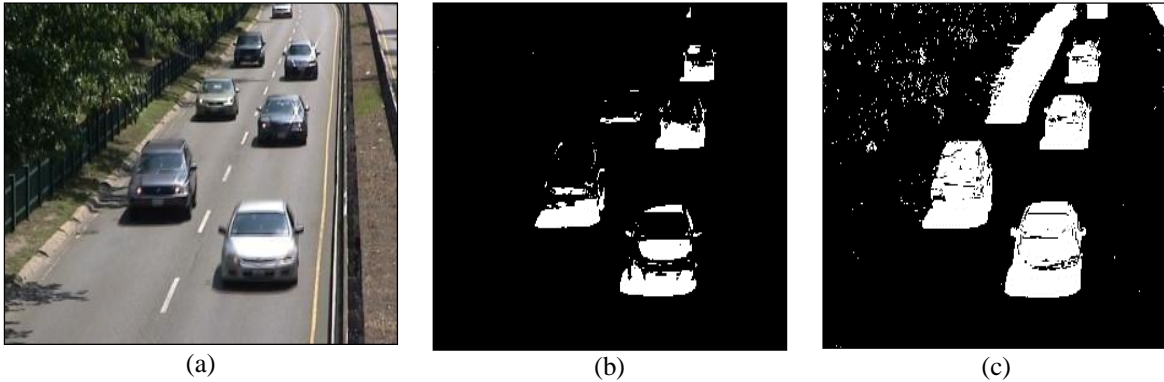


Figure 2. Image result of KDE method using a fixed threshold. (a) Original image, (b) Image result obtained using $P(x) > \text{threshold} = 0.8$, (c) Image result obtained using $P(x) > \text{threshold} = 0.2$.

Assume that X is conditionally independent given Y , then:

$$P(X / Y) = \prod_{p=1}^N P(x_p / y_p) \tag{8}$$

Where

$$P(x_p / y_p) = \exp(-D_p^{obs}(y_p)) \tag{9}$$

$D_p^{obs}(y_p)$ is the data energy, it tests how the current labeling y_p is coherent with the observed data.

Using (8) and (9) we can write $P(X / Y)$ as follows:

$$P(X / Y) = \exp(-\sum_{p=1}^N D_p^{obs}(y_p)) \tag{10}$$

Whereas $P(Y)$ is a prior probability calculated by Hammersley-Clifford theorem that simulates Markov Random Field using Gibbs distribution with a four-neighborhood system as follows:

$$P(Y) = \exp\left(-\sum_{p,q} V_{p,q}(y_p, y_q)\right) \tag{11}$$

$V_{p,q}(y_p, y_q)$ denotes the smooth energy, it penalizes two neighboring pixels p and q when the label y_p and y_q are too different.

Using (10) and (11) we can write:

$$\operatorname{argmax}_{y_p \in Y} P(Y / X) = \operatorname{argmax}_{y_p \in Y} \exp\left(-\sum_{p=1}^N D_p^{obs}(y_p) - \sum_{p,q} V_{p,q}(y_p, y_q)\right) \quad (12)$$

So finally, maximizing the posterior probability $P(Y / X)$ is equivalent to minimizing the energy function E represented as follows:

$$E(y) = \sum_{p \in I} D_p^{obs}(y_p) + \sum_{p,q} V_{p,q}(y_p, y_q) \quad (13)$$

Where, the minimization of the energy function E is equivalent to find a minimum cut in a graph $G = (V; E')$.

3.2.2 Moving Object Detection by Minimization of Energy Function

Our aim is to segment our probability image into background/moving object, this segmentation is estimated as a global minimum of the energy function E (13) computed by a standard minimum cut algorithm[24]-[25], as demonstrate in Figure 3.

Firstly, we create a graph $G = (V; E')$ for each probability image, where V stands for the set of nodes (pixels), and two terminal nodes connected to every pixels, named S (Source) and T (Sink), E' represents the set of edges connecting two adjacent nodes with weight $V_{p,q}(y_p, y_q), (p, q \in N)$, and connecting S and T to every nodes p in I with weight $D_p^{Background}(y_p)$ and $D_p^{Object}(y_p)$ respectively.

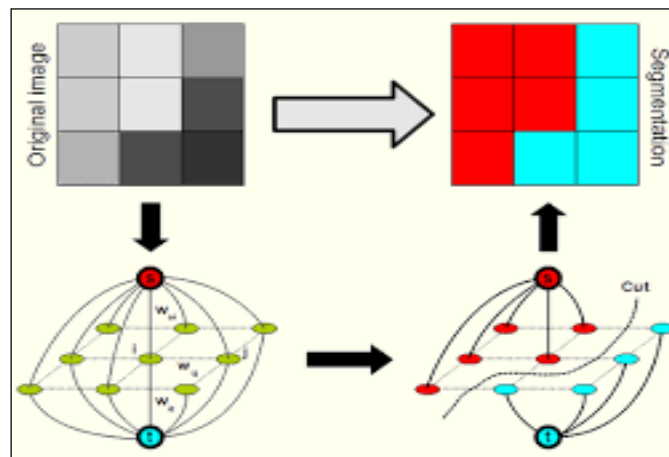


Figure 3. Image segmentation using Graph Cut [24]

Therefore, the binary labeling problem is to assign each node in V a unique label y_p , where y_p in $\{0,1\}$. Secondly, Once the graph is built, Boykov and Jolly[24] solve the labeling of the pixels through a cut on the graph as shown in Figure 3, this cut will sever two types of links:

- a) T-links: a cut removes one of the two edges that connect a pixel with a terminal S or T node, associating it to the object or background class.
- b) N-links: a cut removes the links between pairs of pixels associated to different terminals.

A cut C in graph is a subset of edges which separates the nodes into two parts; one part belongs to the source and the other belongs to the sink. The cost of a cut is the summation of the weights of its edges. The minimum cut of the graph will generate an optimal segmentation in the image.

In order to calculate the energy function E , we calculate $D_p^{obs}(y_p)$ and $V_{p,q}(y_p, y_q)$ as follows:

$$D_p^{Background}(y_p) = e^{-\frac{(x_p - \mu_B)^2}{2\sigma_B^2}} \quad (14)$$

$$D_p^{Object}(y_p) = e^{-\frac{(x_p - \mu_{Obj})^2}{2\sigma_{Obj}^2}} \tag{15}$$

$$V_{p,q}(y_p, y_q) = \frac{1}{dist(p, q)} e^{-\frac{(x_p - x_q)^2}{2\sigma^2}} \tag{16}$$

Where

μ_B : Represents the peak of background distribution histogram represented in Figure 4(b)).

μ_{Obj} : Represents the peak of objects distribution histogram represented in Figure 4(b)).

σ_B and σ_{Obj} : Represents the variance of background and objects distributions respectively.

x_p, x_q : Pixels values in probability image

$Dist(p, q)$: Euclidian distance between pixels p and q .

As shown in Figure 4(b), the histogram of probability image is bimodal. μ_{Obj} is the mean of the first distribution, and μ_B is the mean of the second distribution. In general, the background pixels is more dominant than object pixels, and the probability value of the background is higher than this of the object, So: $\mu_B > \mu_{Obj}$.

μ_{Obj}, μ_B and σ_{Obj}, σ_B are calculated using K means algorithm applied in probability image, with the number of cluster is set to 2.

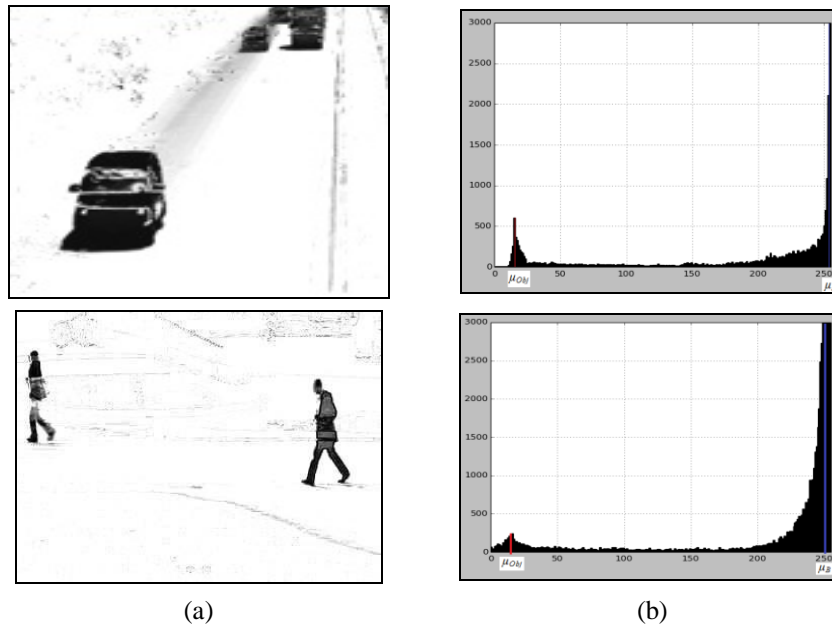


Figure 4. (a) Probability image, (b) Histogram of probability image

4. RESULTS AND ANALYSIS

In order to validate the proposed technique using qualitative evaluation, three different types of scenes were considered from the CDnet2014 video dataset, which are acquired using a fixed camera. The first one is a highway from Baseline sequences which represents a road containing vehicles with non stationary background due to moving trees in left of the image; the size of frame is 320×240 . And the second is a pedestrians from baseline sequences, it is appropriate for testing the effect of illumination change, the size of the frame is 360×240 . Whilst, the third is the fountain1 from Dynamic background sequences, which represents a scenario of dynamic background due to the presence of moving trees and fountain; the size of the frame is 432×288 .

To verify the performance of the proposed method, we compare our results to those obtained using Gaussian Mixture Model (GMM) [5] and Kernel Density Estimator method (KDE) [10]. The qualitative results are illustrated in Figure 5, where the origin frames of the video sequences are shown in the first column, the results obtained using Gaussian Mixture Model are displayed in the second column, and those of

KDE method are shown in the third column, those acquired with the proposed method in the fourth column, and the fifth column represents the ground truth images.

The implementation of the proposed and the comparison methods is developed using Opencv 2.4.9 in C++ environment, where the GMM is tested by using the number of Gaussians=3, the number of historic frames=100 and learning rate $\alpha=0.2$. Whereas KDE method is implemented by using the number frames in buffer=100, learning rate $\alpha=0.2$ and threshold=0.6.=

As illustrated in Figure 5, the results of the first sequence indicate that all methods successfully detected the foreground, which may be explained by the fact that the sequence is simple. However, the proposed method shows more effectiveness in suppression the false positive and negative pixels. In the second sequence, each method presents a degree of success to cope with uninteresting movements of the background due to moving trees and fountain. However the results of the GMM and KDE contain not only missed foreground detection but also false foreground detection. But the use of KDE and Graph cut has improved the quality of detection, especially, the correction of misclassified foreground and background pixels. Whereas the results of the third sequence show that our method has effectively adapted to the illumination change of the scene.

Whereas in the quantitative analysis, three different performance metrics, Precision, Recall and F-Measure have been tested, and are defined as follows:

$$\text{Precision} = \frac{TP}{TP + FP} = \frac{\text{Number of true positives detected}}{\text{Number of positives detected}}$$

$$\text{Recall} = \frac{TP}{TP + FN} = \frac{\text{Number of true positives detected}}{\text{Total number of true positives in ground truth}}$$

$$F - \text{Measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Where TP denotes true positives, FP represents false positives and FN false negatives.

Among these metrics, we are interested especially in the F-measure, which are commonly accepted as good indicators of the overall performance of the background subtraction methods. By definition, a good algorithm is the one that produces a small number of false positives, false negatives, and high F-measure score.

The average recall, average precision and average F-measure values of each method are calculated using all baseline and dynamic background sequences and illustrated in Table 1, which reveals that the proposed scheme produces better F-measure score compared to the others approaches, either in the baseline and dynamic background sequences, except KDE, which presents a better score in the baseline, but our method performs well in the dynamic background sequence.

Table 1. The Average Metrics Score of the Proposed and other Methods, Note that These Results Obtained from CDnet 2014 Challenge Website

Sequences	Average Metrics	GMM[5]	KDE[10]	MST BG Model [11]	DCB [15]	Proposed Method
Baseline	Precision	0.8461	0.9223	0.8870	0.9070	0.9011
	Recall	0.8180	0.8969	0.8137	0.7123	0.8465
	F-Measure	0.8245	0.9092	0.8450	0.7695	0.8729
Dynamic Background	Precision	0.5989	0.5732	0.5515	0.7632	0.6021
	Recall	0.8344	0.8012	0.7392	0.5803	0.6815
	F-Measure	0.6330	0.5961	0.5953	0.6149	0.6393

As demonstrated above, the combination of KDE and Graph Cut methods improves the quality of results as shown in comparative results, because, it exploits the correlation that exists between the intensities of neighboring pixels, which achieve higher detection accuracy in the presence of dynamic background.

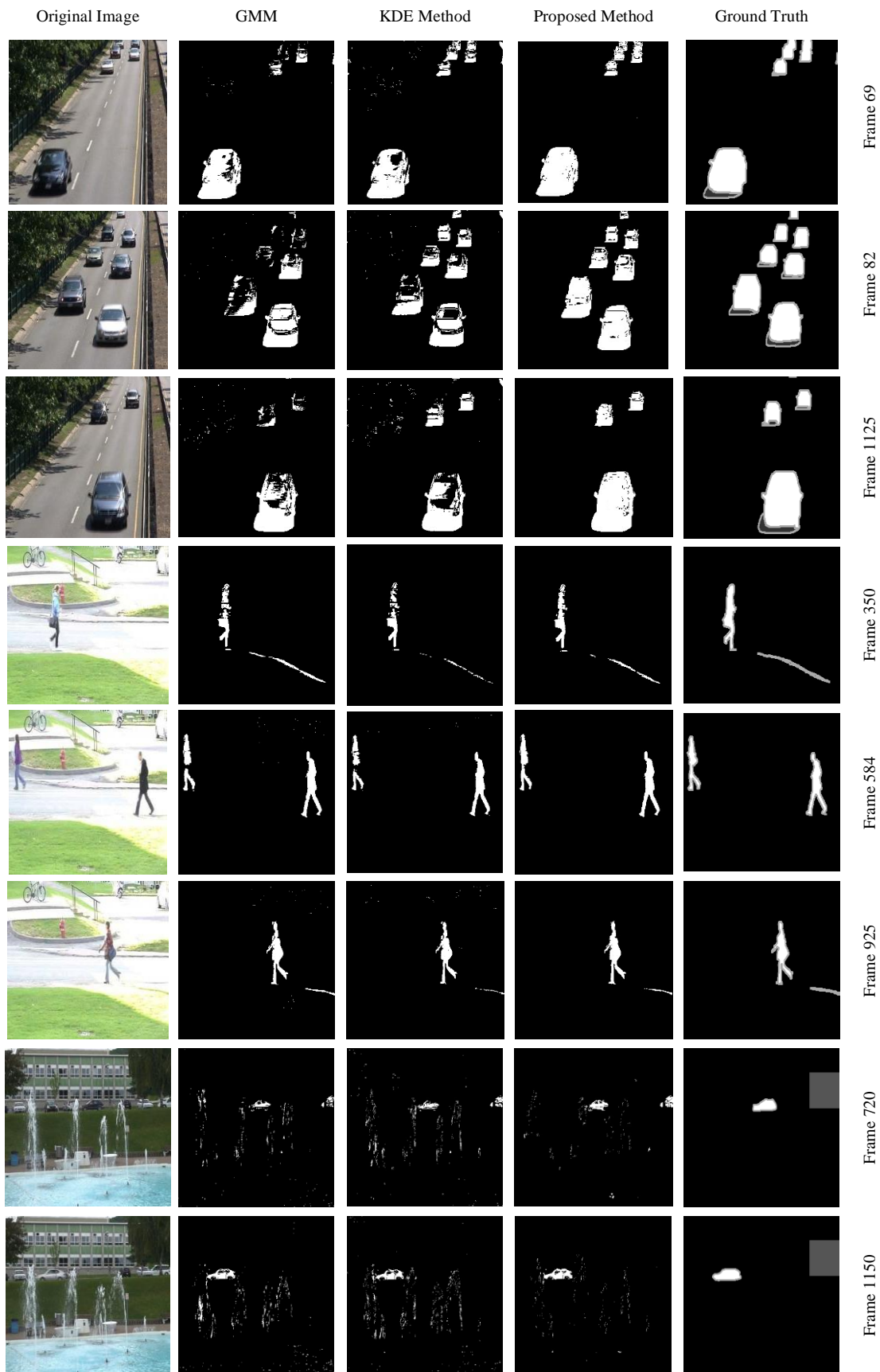


Figure 5. Results of the proposed, GMM and KDE methods

5. CONCLUSION

The purpose of the current paper is to detect moving objects from the background, by tackling two issues, exploiting the correlation between neighboring pixels and avoid the use a global threshold. So firstly we generate a probability image using Kernel Density Estimation method, secondly, instead of threshold the probability image to extract moving pixels, this task is achieved by using the graph cut algorithm, which minimize an energy function, that is equivalent to classifying each pixel into the background or the foreground. This approach has been tested in CDnet2014 video dataset, and the obtained results demonstrate its effectiveness in foreground detection with less memory requirement and time consuming, compared to the others methods.

REFERENCES

- [1] Satish Kumar N, and G. Shobha. Background Modeling to Detect Foreground Objects Based on ANN and Spatio-Temporal Analysis. *Indonesian Journal of Electrical Engineering and Computer Science* 2017; 15(4): 151-160.
- [2] Xu Y, Dong J, Zhang B, and Xu D. Background modeling methods in video analysis: A review and comparative evaluation. *CAAI Transactions on Intelligence Technology* 2016; 1(1): 43-60.
- [3] Yao G, Lei T, Zhong J, Jiang P, and Jia W. Comparative Evaluation of Background Subtraction Algorithms in Remote Scene Videos Captured by MWIR Sensors. *Sensors* 2017; 17: 1945.
- [4] Thierry B, Sajid J, Maryam S, and Soon Ki J. Deep Neural Network Concepts for Background Subtraction: A Systematic Review and Comparative Evaluation. *CoRR*. 2018; abs/1811.05255.
- [5] Stauffer C and Grimson W. E. L. Adaptive background mixture models for real-time tracking. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition 1999; Cat. No PR00149*.
- [6] Zivkovic Z. Improved Adaptive Gaussian Mixture Model for Background Subtraction. *17th International Conference on Pattern Recognition (ICPR)* 2004.
- [7] Xia, Haiying, Song S, and He L. A modified Gaussian mixture background model via spatiotemporal distribution with shadow detection. *Journal of Signal, Image and Video Processing* 2015; 10(2): 343-350.
- [8] Mourad M, Maha H and Ali D. Comparative study of statistical background modeling and subtraction. *Indonesian Journal of Electrical Engineering and Computer Science*, 2017; 8(2): 287–295.
- [9] Boutaina H, Mohammed El Haj T, Rachid Oulad Haj T and Rdouan F. Detecting and Shadows in the HSV Color Space using Dynamic Thresholds,” *International Journal of Electrical and Computer Engineering*, 2018; 8(3): 1513-1521.
- [10] Elgammal A, Harwood D and Davis L. Non-parametric model for background subtraction. *European Conference on Computer Vision. Springer Berlin Heidelberg* 2000.
- [11] Park J.G and Lee C. Bayesian rule-based complex background modeling and foreground detection. *Optical Engineering* 2010; 49(2).
- [12] Xiqun L. A multiscale spatio-temporal background model for motion detection”, *IEEE International Conference on Image Processing (ICIP)* 2014.
- [13] Lee J. and Park M. An adaptive background subtraction method based on kernel density estimation. *Sensors (Basel, Switzerland)* 2012; 12(9): 12279–12300.
- [14] Shah M, Deng J.D, and Woodford B.J. A Self-adaptive CodeBook (SACB) model for real-time background subtraction. *Image and Vision Computing* 2015; 38: 52–64.
- [15] Rawitas K and Worapan K. Foreground segmentation in a video by using a novel dynamic codebook. *13th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)* 2016.
- [16] Heikkila M and Pietikainen M. A texture-based method for modeling the background and detecting moving objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 2006; 28(4): 657–662.
- [17] Zhao G and Pietikaine M. Dynamic texture recognition using local binary patterns with an application to facial expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 2007; 29(6): 915–928.
- [18] Zhang S, Yao H, and Liu S. Dynamic background modeling and subtraction using spatio-temporal local binary patterns. *15th IEEE International Conference on Image Processing (ICIP)* 2008.
- [19] Bilodeau G. A, Jodoin J. P, and Saunier N. Change detection in feature space using local binary similarity patterns. *International Conference on Computer & Robot Vision* 2013; 10(1).
- [20] Lim Long A, Hacer Yalim K. "Foreground Segmentation Using Convolutional Neural Networks for Multiscale Feature Encoding. *Pattern Recognition Letters* 2018.
- [21] Wang Y, Luo Z. M, and Jodoin P. M. Interactive Deep Learning Method for Segmenting Moving Objects *Pattern Recognition Letter* 2016.
- [22] Vijayan M and Mohan R. Moving object detection using vector image model. *Optik* 2018; 168: 963-973.
- [23] Mohammadreza B, Tung Dinh D, and Rigoll G. A Deep Convolutional Neural Network for Video Sequence Background Subtraction. *Pattern Recognition* 2017.
- [24] Yuri Y. B, Marie-pierre J. Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images. *IEEE International Conference on ComputerVision* 2001.
- [25] Khalid H, Driss M and Youssef C. Interactive ROI Segmentation using Graph Cuts. *International Journal on Graphics, Vision and ImageProcessing* 2009; 9(6): 1-6.