2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

# Real-time Background Subtraction under Varying Lighting Conditions

Sisi Liang[1] and Darren Baker[1]

*Abstract*— **Background subtraction is an important topic in computer vision and video analysis. It is challenging to robustly segment foreground and background in complex scenarios. In the literature there are efforts to address some of the main challenges such as illumination change, dynamic backgrounds, hard shadows, and intermittent object motion. However, most of the research has focused on applying advanced mathematical and machine learning models rather than on improving performance in real-time applications. In this paper, we devise a method named EGMM to efficiently handle the illumination change problem and also operate at a real-time execution speed on commodity PC hardware. EGMM is an ensemble algorithm that fuses multiple Gaussian Mixture Models operating on gradient, texture and color features. Detection and removal of shadows is done using a chromaticity-based approach, and spatio-temporal history of foreground blobs is used to handle intermittent object motion. We benchmarked EGMM by creating datasets for two light change scenarios. The results demonstrate that EGMM achieves robust performance in complex illumination change cases, outperforms some state-of-the-art algorithms, and runs at** $100$ **fps (GPU) at** $1280 \times 720$ **resolution. Moreover, experiments using the 2012 CDnet dataset show that EGMM achieves generally good performance in varying scenes with overall results better than conventional methods and runs at** $1000$ **fps (GPU) at** $320 \times 240$ **resolution.**

## I. INTRODUCTION

Background subtraction is a fundamental video processing task used in a variety of computer vision applications. These include intelligent video surveillance, human activity recognition, and industrial vision [1]. Background subtraction aims to segment an input video frame into foreground and background regions, where the foreground regions represent moving objects and the background regions represent a static scene. However, accurate background subtraction for complex scenes in real time is a difficult task due to the complexities of real-world scenarios. Some of the key challenges are: illumination changes (e.g switching a light on or off), dynamic background (e.g swaying trees, camera jitter), and shadows.

Over the years, there have been efforts to improve background subtraction methods in complex scenes. Background subtraction methods can be loosely grouped into two categories: (1) unsupervised algorithms, and (2) supervised algorithms. Unsupervised algorithms refers to methods that mathematically model the background and then predict the foreground accordingly. In this category, the Gaussian Mixture Model (GMM) [2]–[5] that relies on pixel intensities to track background representation is one of the most popular methods. Supervised algorithms are machine learning methods which aim to learn the parameters of a complex function in order to minimize a loss function of the labeled training frames [6]. Supervised algorithms using deep neural networks (DNNs) [6]–[8] have been recently applied to background subtraction and show better results than traditional unsupervised methods. Most background subtraction methods presented in the literature tend to focus on development of advanced mathematical models and machine learning models, while research targeting both real-time and robust performance in challenging real-world situations is comparatively neglected [1]. We contend that this line of work is similarly important since a major use of background subtraction is in real-time applications such as video surveillance.

Here we propose a background subtraction method with robust performance under varying illumination and with high processing speed. Our proposed method is an ensemble algorithm that fuses multiple instances of GMM and is based on the work of [9]. We made several improvements to [9] as explained in the following. First, we used modified GMM2 [3] (see section III-D) instead of the MOG [10] method used by [9] to better cope with illumination change and intermittent objects. Second, we conducted a feature selection process to choose 9 optimal features from the 13 features used by [9]. Third, we applied a two-level fusion method for the GMM ensemble to achieve consistent performance versus a single-level fusion method used by [9]. Fourth, we added chromaticity-based shadow detection to handle large moving shadows. Fifth, the ensemble algorithm in [9] is not a real-time classifier. Our method is efficiently implemented on GPU to achieve real-time execution speed. To our knowledge, there is no public dataset that provides both videos showing large illumination changes and enough pixel-wise labels (eg. some datasets have only one manually labelled frame for a video, others provide only object bounding boxes). To test our method, we created a dataset with over 2000 manual annotations (see section IV-A) and the dataset download link is https://doi.org/10.25919/f1pm-rq13. Our contributions are summarized as follows:

- We designed an efficient background subtraction method with the use of different image features in an ensemble of GMMs, significantly improving performance in video sequences with strong illumination changes;
- We implemented the proposed method on GPU to achieve real-time high-resolution image processing;
- We developed a chromaticity-based shadow detection method to improve foreground segmentation;
- We utilized spatial and temporal history of foreground

[1]Sisi Liang and Darren Baker are with the Robotics and Autonomous Systems Group, Data61, CSIRO, Brisbane, QLD 4069, Australia firstname.lastname@data61.csiro.au

blobs to handle intermittent object motion;
- We created a new benchmark dataset for evaluation of background subtraction methods under large illumination changes.

## II. RELATED WORK

Early attempts at background subtraction first compute a statistical background model and then use it to segment the foreground. Background subtraction using a probabilistic GMM is a widely used approach. In the GMM algorithm, each pixel intensity is characterized by multiple, weighted, Gaussian distributions. A GMM was originally proposed for background subtraction in [11] and an efficient update was then given in [2]. Many improvements of GMM have been made since then. In [3], Zivkovic presented an improved GMM algorithm, namely GMM2, by creating an online procedure for the update of GMM parameters to automatically adapt to the scene. In [4], Chen et al. proposed a self-adaptive GMM that used a dynamic learning rate with adaptation to global illumination to cope with illumination changes. In [5], Shah et al. introduced a local parameter learning algorithm for the GMM and used a SURF (speeded up robust features) feature matching algorithm to suppress ghosts in the foreground mask caused by illumination changes. Other methods have also been developed using texture features in the form of local binary similarity pattern (LBSP) [12]. The advantage of these methods is that texture features provide richer background information than color information alone. St-Charles et al. [13] proposed a method which adapted and integrated LBSP features to the non-parametric ViBe method [14]. Based on their previous work, they then developed SUBSENSE [15] and PAWCS [16] models by incorporating more features (color, persistence indicators) to improve foreground detection and ignore disturbances caused by illumination variations. In [17], Lee et al. proposed a WisenetMD model which was in part derived from a SUBSENSE model with a particular focus on removing false positives in the dynamic background region by adding a recheck module.

In recent years, DNNs have been widely applied to background subtraction. Various authors [7], [18], [19] have employed a convolutional neural network (CNN) for background subtraction. Other authors proposed variants of CNN architecture such as a triplet CNN [8], a 3D CNN [20]. Through another study, Bakkayet al. [21] used generative adversarial networks for background subtraction. Although all these algorithms perform very well on different public background subtraction datasets, it is important to note that they use some frames from test videos for model training and all frames of the same videos for testing, thus they will suffer a performance loss when tested on unseen videos. Recently, several supervised methods for unseen videos have been introduced. Kim and Ha [22] adopted a U-Net architecture [23] which took a background model image and multiple original images as input. They divided a background subtraction dataset into a training set and a test set, and evaluated the algorithm on the test videos that were unseen during the training. Tezcan et al. [6], [24] also used a U-Net with the addition of a novel augmentation step to address different challenges (e.g illumination change, camera jitter), thus improving performance on unseen videos. Although these deep learning based methods can be naturally considered for being robust in the various challenges met in videos, they are still too time and memory consuming to be used for real-time applications. In addition, these methods need manual annotation of data to train the DNNs and are often scene-specific. To find a good trade-off between accuracy and execution speed, in this paper, we focus on building a practically fast background subtraction system with good segmentation accuracy, particularly to address the illumination change challenge.

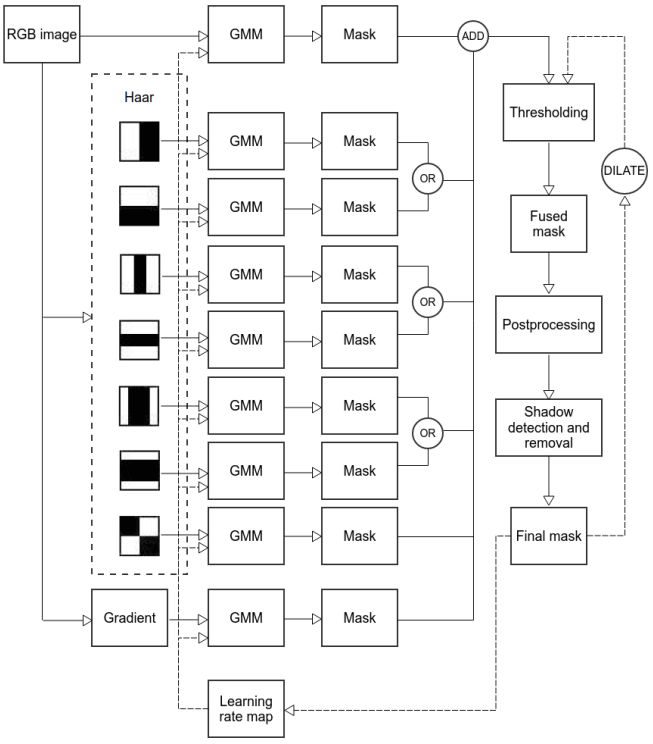


Fig. 1. The framework of EGMM. The final mask is used as a reference to control the learning rate where the foreground region applies lower learning rate and the background region applies higher learning rate. Low and high thresholds respectively are applied to the foreground and background regions of the dilated final mask during thresholding.

## III. PROPOSED METHOD

The framework of our method, coined EGMM (Ensemble of GMMs), is shown in Figure 1. Specifically, we extract color, texture and gradient features, let a GMM operate on each individual feature, and then apply a hierarchical fusion method, a postprocessing step, and shadow detection and removal. The general idea behind this method is to use some image features that are less susceptible to varying illumination and let each GMM classifier operate exclusively on one of these image features. By operating on different features, this method demonstrates robust performance on video frames with strong illumination changes.

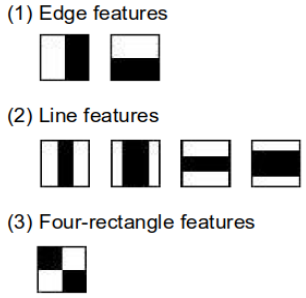


Fig. 2. The set of Haar-like features being used.

### A. Feature Extraction

The first step in the proposed method is to extract distinct features from an image. Nine different features are used. The first feature is simply color intensity (values of RGB). RGB feature focuses on the pixel and ensures precision of the pixel observation.

The next seven texture features are Haar-like features [25], which are calculated by summing the pixel intensities in each region and subtracting the sum of a region from the sum of the remaining region. Haar-like features have been successfully used for face detection [26], [27]. Quick calculation of Haar-like features can be achieved by using an integral image [25].

In this paper, we use seven Haar-like feature prototypes including two edge features, four line features, and one four-rectangle feature as shown in Figure 2. The edge features are responsible for detecting edges in a horizontal or in a vertical direction. The line features are responsible for finding lines (i.e. a lighter region surrounded by a darker region on either side and vice-versa) in a horizontal or in a vertical direction. The four-rectangle features are responsible for finding diagonal lines. Every Haar-like feature is computed in a $12 \times 12$ region surrounding each pixel.

The final feature is the image gradient. A Sobel edge detector [28] is used with a $3 \times 3$ kernel. The intensity gradients in the x and y directions are:

$$G_{\mathrm{x}} = A * I \ , \tag{1}$$

$$G_{\mathrm{y}} = B * I \ , \tag{2}$$

where $A$ and $B$ denote the x-direction and y-direction kernels respectively, and $I$ represents the input image. The gradient magnitude $G$ can be then computed as:

$$G = \sqrt{{G_{\mathrm{x}}}^2 + {G_{\mathrm{y}}}^2} \ . \tag{3}$$

The gradient magnitude and Haar-like features for a pixel are calculated on the relationship to neighboring pixels. If each pixel in a neighborhood undergoes the same change, then ideally the difference between neighboring pixels remains constant. During an illumination change, neighboring pixels often experience the same change in illumination. Thus features like gradient magnitude and Haar-like features can remain unchanged under varying lighting.

### B. Ensemble Algorithm

After the nine features are extracted, each GMM classifier operates only on one of these features. The proposed ensemble algorithm is to employ a two-level fusion method to fuse the predictions of multiple classifiers as shown in Figure 1. The motivation for using a two-level fusion arose from the inconsistent performance of a single-level fusion. In our experiments, choosing a global threshold for fusing predictions of nine classifiers was found difficult to achieve consistent performance across our data set. In a simple single-level fusion method, the prediction of each classifier is equally weighted. However, some features contain more information than others among these nine features. For example, the image gradient incorporates information in both x and y directions, whereas edge and line features in either x or y direction only extract information in one direction. To balance the information each feature contains, the predictions of edge and line features in a horizontal direction direction are joined with those in a vertical direction by using the Boolean operator OR to obtain the enriched results. Alternatively, we can employ weighted voting which needs to select appropriate weights for all the classes per classifier and could involve using the optimization algorithm and a holdout dataset [29], [30]. Our experiments show that fusion by merging certain complementary feature results are robust enough to cope with illumination changes. Then the first level fusion results and the outputs of classifiers on the remaining features are added together in the second level fusion process.

Subsequently, the majority vote [31] is applied to the sum map to determine the class label for each pixel in the fused mask. Let $H$ denote a binary classifier, and its output is either background (0) or foreground (1). Equations 4 and 5 show how the hypothesis $H_{\mathrm{i,j}}$ for each pixel $p_{\mathrm{i,j}}$ is derived using the majority vote.

$$sum = \sum_{k=1}^{6} H_{\mathrm{i,j}}(k) \ , \tag{4}$$

$$H_{\mathrm{i,j}} = \begin{cases} 1 & sum > \tau \\ 0 & otherwise \end{cases} , \tag{5}$$

where $H_{\mathrm{i,j}}(k)$ represents the hypothesis generated from the $kth$ feature for a pixel at coordinates $(i, j)$. The per-pixel hypothesis $H_{\mathrm{i,j}}$ is considered a foreground class if $sum$ is larger than the threshold $\tau$, otherwise it is considered a background class. We use low and high $\tau$ for foreground and background regions of the dilated mask respectively.

Post-processing is then applied to the fused mask to further remove the noise and fill object holes by using a series of image processing operations including morphological processing, box filtering, binary thresholding, and flood filling. More details can be found in the supplementary video.

### C. Shadow Detection

GMMs tend to misclassify moving cast shadows as the foreground since shadows have the same movement patterns

and a similar magnitude of intensity change as that of the foreground objects [32]. As such, shadow detection and removal become an unavoidable step for improving foreground object detection. We chose the HSV (Hue-Saturation-Value) approach proposed by Cucchiara et al. [33] for shadow detection since it was easy to implement and computationally inexpensive. The main reason for using the HSV color space rather than the RGB color space is that it can better separate between chromaticity and luminosity and have proven easier to set a mathematical formulation for shadow detection [33]. The shadow mask $S$ for each pixel $(i, j)$ is defined with three conditions as follows:

$$S(i,j) = \begin{cases} 1 & \alpha < \frac{I^V(i,j)}{B^V(i,j)} < \beta \\ & \wedge\, I^S(i,j) - B^S(i,j) < \tau_{\mathrm{S}} \\ & \wedge\, |I^H(i,j) - B^H(i,j)| < \tau_{\mathrm{H}} \\ 0 & otherwise \end{cases} \tag{6}$$

where $I$ and $B$ denote an input image and a background image respectively, and $\alpha, \beta \in [0, 1]$. The lower bound $\alpha$ accounts for how strong the light source is and defines a maximum value for the darkening effect of shadows on the background. The upper bound $\beta$ is used to avoid identifying the points where the background was slightly changed by noise as shadows. The experiments showed that the difference in saturation between $I$ and $B$ was usually negative for shadow points [34]. Hence the upper threshold $\tau_{\mathrm{S}}$ is performed on the difference of $S$. According to experimental tests, a threshold on the absolute difference of $H$ produced better results [34]. The choice of parameters $\alpha$, $\beta$, $\tau_{\mathrm{S}}$ and $\tau_{\mathrm{H}}$ is done empirically in this work and is based on the average $H$, $S$ and $V$ over the pre-shadow-removal foreground mask that can be measured directly.

### D. Intermittent Objects

The presence of intermittently moving objects is another issue that confounds traditional background models including GMM-based methods. Once foreground objects stop moving for a certain amount of time, they will be incorporated into the background.

In the original GMM2 method implemented in this work, a global learning rate $\alpha \in [0, 1]$ is applied to an input image to update the background model. When the foreground object stops and remains static for some time it will be temporally presented as an additional cluster or Gaussian component. If the object remains static long enough, the weight of the new cluster becomes larger than the predefined threshold and it is then absorbed into the background. To address this issue, we use the spatial and temporal history of the foreground mask as a reference to control the learning rate in different regions as shown in Figure 1. In the foreground region, a small value of the learning rate is applied in order to slow down the weight increase of the new cluster generated by the paused object. In the background region, a relatively higher learning rate is used to quickly adapt to the scene. We apply low $\alpha = 0.0005$ and high $\alpha = 0.005/0.008$ by default and may adjust their values on a case-by-case basis.

## IV. EXPERIMENTS

### A. Datasets

We empirically searched for public light change datasets that could provide videos showing large illumination changes together with enough pixel-wise annotations. However, we did not find such datasets that meet these two criteria. In order to examine how well EGMM performs in varying lighting conditions, we created two new light change datasets:

Light change scenario 1 dataset: Two video sequences, each captured by a different camera. A person walks towards a rolling door, the door gradually opens up causing illumination changes, and then the person continues walking near the door. Each video has 1500 frames including 501 manually labelled frames.

Light change scenario 2 dataset: Two video sequences, each captured by a different camera. A person walks, switches off the light, keeps on walking, switches on the light, and then continues walking. Each video has 3100 frames including 502 manually labelled frames.

The spatial resolution of videos in both light change datasets is $1280 \times 720$.

In order to evaluate the general performance of EGMM in other scenarios, we used the 2012 CDnet dataset [35]. This dataset is one of the most frequently used datasets for benchmarking change detection algorithms and contains a total of 31 video sequences spanning six categories: baseline, camera jitter, dynamic background, intermittent object motion, shadow, and thermal. The spatial resolution of videos ranges from $320 \times 240$ to $720 \times 576$ pixels. Every video contains from 350 up to 6100 ground truth frames which are labeled pixel-wise as follows: 1) foreground, 2) background, 3) hard shadow, 4) unknown motion, or 5) non-ROI (region of interest).

### B. Evaluation Metrics

For quantitative analysis, we use seven standard performance metrics provided by the 2012 CDnet dataset, namely recall (Re), specificity (Sp), false positive rate (FPR), false negative rate (FNR), percentage of wrong classifications (PWC), precision (Pr) and F-measure(FM). Let $TP$ = number of true positives, $TN$ = number of true negatives, $FN$ = number of false negatives, and $FP$ = number of false positives. These seven metrics are defined as follows: Re = $TP/(TP + FN)$, Sp = $TN/(TN + FP)$, FPR = $FP/(FP + TN)$, FNR = $FN/(TP + FN)$, PWC = $100 * (FN+FP)/(TP+FN+FP+TN)$, Pr = $TP/(TP+FP)$, FM = $(2 * Pr * Re)/(Pr + Re)$.

### C. Results

Table I presents quantitative results of EGMM and other previous models for light change scenario 1 dataset. EGMM gives near-optimal results while all four previous methods struggle to handle large illumination changes present in this dataset, and as a consequence, they produced more false positives which significantly dragged down their Pr scores. Qualitative results of PAWCS, GMM2 and EGMM are shown in Figure 3.

TABLE I

RESULTS OF EGMM AND OTHER METHODS TESTED ON LIGHT CHANGE SCENARIO 1 DATASET.

| Method | Re | Sp | FPR | FNR | PWC | Pr | FM |
|---|---|---|---|---|---|---|---|
| **EGMM** | 0.868 | 0.998 | 0.002 | 0.132 | 0.292 | 0.730 | **0.791** |
| PAWCS [16] | 0.967 | 0.959 | 0.0406 | 0.033 | 4.056 | 0.144 | 0.246 |
| SUBSENSE [15] | 0.843 | 0.990 | 0.010 | 0.157 | 1.068 | 0.388 | 0.505 |
| LOBSTER [13] | 0.590 | 0.965 | 0.035 | 0.410 | 3.695 | 0.110 | 0.184 |
| GMM2 [3] | 0.394 | 0.990 | 0.010 | 0.606 | 1.420 | 0.262 | 0.289 |

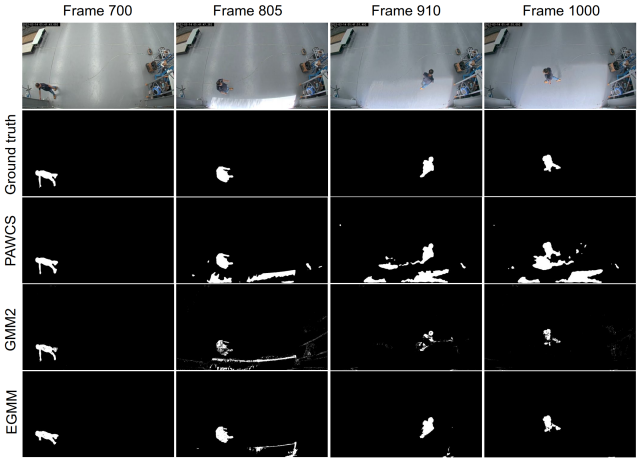


Fig. 3. Comparative results of PAWCS, GMM2, and EGMM methods tested in light change scenario 1 dataset. The columns correspond to the frame number in the video sequence and the rows show the ground truth and the background subtraction results.

In Table II, we compare EGMM against the same four methods for light change scenario 2 dataset. Here we note that all models except for GMM2 perform satisfactorily in this dataset. Both EGMM and PAWCS achieved similarly high F-measure scores and were the top ranked models in Table II. Figure 4 shows the comparative results of PAWCS, GMM2 and EGMM models tested on the scenario 2 dataset.

TABLE II

RESULTS OF EGMM AND OTHER METHODS TESTED ON LIGHT CHANGE SCENARIO 2 DATASET.

| Method | Re | Sp | FPR | FNR | PWC | Pr | FM |
|---|---|---|---|---|---|---|---|
| **EGMM** | 0.900 | 0.999 | 0.0004 | 0.099 | 0.134 | 0.944 | **0.921** |
| PAWCS [16] | 0.919 | 0.999 | 0.0006 | 0.080 | 0.131 | 0.929 | 0.924 |
| SUBSENSE [15] | 0.887 | 0.998 | 0.002 | 0.113 | 0.284 | 0.813 | 0.846 |
| LOBSTER [13] | 0.835 | 0.999 | 0.001 | 0.165 | 0.206 | 0.923 | 0.876 |
| GMM2 [3] | 0.743 | 0.972 | 0.028 | 0.257 | 2.993 | 0.190 | 0.302 |

Considering results in both Table I and Table II, it is clear that EGMM achieves robust performance in two challenging light change datasets. Scenario 1 dataset is more difficult than scenario 2 since it not only involves light changes but also a dynamic background (eg. reflections on the ground, a rolling door). PAWCS, SUBSENSE, LOBSTER methods perform poorly in scenario 1 but well in scenario 2, indicating that they have the capability to cope with certain illumination changes such as indoor light changes but may still suffer under more complex light changes.

Next, in Table III, we compare the processing speed

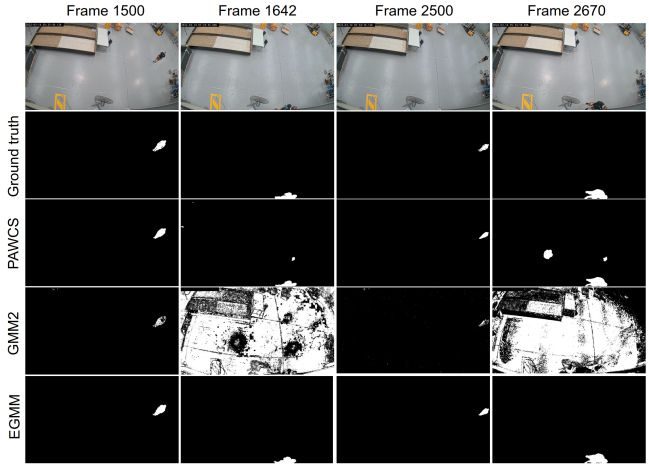


Fig. 4. Comparative results of PAWCS, GMM2, and EGMM methods tested in light change scenario 2 dataset. The columns correspond to the frames number in the video sequence and the rows show the ground truth and the background subtraction results.

of EGMM, GMM2, PAWCS, SUBSENSE and LOBSTER methods tested on light change datasets. C++ implementations of all five methods based on OpenCV are used in this work. Here EGMM and GMM2 both have GPU implementations while others do not. As shown in the table, PAWCS, SUBSENSE and LOBSTER methods appear much slower than EGMM and GMM2 for processing higher resolution images. EGMM has a fast processing speed of 100 fps at $1280 \times 720$ resolution on a Nvidia GeForce GTX 1080 GPU and gives outstanding results (Tables I and II), thus making it very useful for real-time applications.

TABLE III

SPEED COMPARISON ON EGMM AND OTHER METHODS TESTED ON LIGHT CHANGE DATASETS. NOTE THAT IF A METHOD HAS A GPU IMPLEMENTATION ITS SPEED ON GPU IS REPORTED.

| Method | Platform | Fps $1280\times720$ |
|---|---|---|
| **EGMM** | PC-i7 3.70 GHz/Nvidia GTX 1080 | **100** |
| GMM2 [3] | Nvidia GTX 1080 | 595 |
| LOBSTER [13] | PC-i7 3.70 GHz | 6 |
| SUBSENSE [15] | PC-i7 3.70 GHz | 3 |
| PAWCS [16] | PC-i7 3.70 GHz | 0.56 |

To compare EGMM with various existing methods in the literature, F-measure is used as the baseline since the general performance of a method is usually closely related to its F-measure score as noted in [35]. Since EGMM is an unsupervised algorithm, comparing it with supervised algorithms such as deep learning-based methods would not be fair as they use a certain amount of ground truth data for model training. Instead, we chose state-of-the-art unsupervised methods including PAWCS, WisenetMD, FBS-ABL [36], and older but popular methods including ViBe+ [37], KNN [38], GMM2 for comparison as shown in Table IV. We can see that EGMM achieves comparable performance to state-of-the-art methods in most categories and outperforms most of classic methods in terms of the

TABLE IV

OVERALL AND PER-CATEGORY F-MEASURE SCORES OF DIFFERENT BACKGROUND SUBTRACTION METHODS AND THE PROPOSED METHOD TESTED ON THE 2012 CDNET DATASET.

| Method | Baseline | Cam. jitter | Dyn. backgr. | Inter. obj. motion | Shadow | Thermal | Overall |
|---|---|---|---|---|---|---|---|
| **EGMM** | 0.899 | 0.778 | 0.753 | 0.647 | 0.869 | 0.799 | **0.791** |
| PAWCS [16] | 0.940 | 0.814 | 0.894 | 0.776 | 0.891 | 0.832 | 0.858 |
| WisenetMD [17] | 0.949 | 0.823 | 0.838 | 0.726 | 0.898 | 0.815 | 0.842 |
| SUBSENSE [15] | 0.950 | 0.815 | 0.818 | 0.657 | 0.899 | 0.817 | 0.826 |
| DPGMM [39] | 0.929 | 0.748 | 0.814 | 0.542 | 0.813 | 0.813 | 0.776 |
| LOBSTER [13] | 0.924 | 0.742 | 0.568 | 0.577 | 0.873 | 0.825 | 0.751 |
| FBS-ABL [36] | 0.865 | 0.530 | 0.742 | 0.723 | 0.867 | 0.662 | 0.732 |
| ViBe+ [37] | 0.871 | 0.754 | 0.720 | 0.509 | 0.815 | 0.665 | 0.722 |
| KNN [38] | 0.841 | 0.689 | 0.686 | 0.503 | 0.747 | 0.604 | 0.679 |
| GMM2 [3] | 0.838 | 0.567 | 0.633 | 0.533 | 0.732 | 0.655 | 0.660 |

overall F-measure score. We modified the GMM2 method to address intermittent object motion. This technique would work best if the learnt background remain static. In some videos, the initial learnt background includes objects such as cars. When these objects later move, the newly revealed background areas, called "ghosts", are detected, resulting in false positives. Adding an additional step such as an object detection component to our method could help identify ghosts and thus improve performance in this case.

Next, in Table V, we compare the speed of EGMM and other methods tested on different platforms. From these results, it is clear that EGMM appears much faster than most other methods including state-of-the-art methods such as PAWCS, Fast BSUV-Net2.0 [24]. EGMM can run at 1000 fps at $320 \times 240$ resolution on a Nvidia GeForce GTX 1080 GPU. Furthermore, GMM2, WisenetMD, FBS-ABL and EGMM have superior execution speed and thus are potentially useful for real-time applications. Among these methods, EGMM outperforms others except WisenetMD in terms of the overall F-measure score. Most background subtraction research focuses on classification performance rather than real-time application performance. The results in Table IV and Table V demonstrate that EGMM can achieve a better trade-off between classification performance and execution speed.

TABLE V

SPEED COMPARISON OF DIFFERENT BACKGROUND SUBTRACTION METHODS AND PLATFORMS TESTED ON THE 2012 CDNET DATASET. NOTE THAT IF A METHOD HAS A GPU IMPLEMENTATION ITS SPEED ON GPU IS REPORTED.

| Method | Platform | Fps 320×240 |
|---|---|---|
| **EGMM** | PC-i7 3.70 GHz/Nvidia GTX 1080 | **1000** |
| GMM2 [3] | Nvidia GTX 1080 | 5434 |
| FBS-ABL [36] | PC-i7 3.3 GHz, 16 GB RAM | 655 |
| KNN [38] | PC-i7 3.70 GHz | 333 |
| WisenetMD [17] | PC-i7 3.60 GHz/Nvidia GTX 1080 | 150 |
| LOBSTER [13] | PC-i7 3.70 GHz | 100 |
| SUBSENSE [15] | PC-i7 3.70 GHz | 62 |
| Fast BSUV-Net2.0 [24] | Nvidia Tesla P100 | 29 |
| DPGMM [39] | Nvidia GTX 580 | 28.5 |
| PAWCS [16] | PC-i7 3.70 GHz | 12 |
| BSUV-Net2.0 [24] | Nvidia Tesla P100 | 6 |

Figure 5 shows the comparative results of PAWCS and EGMM on "pedestrians", "bugalows", "busstation" video sequences where large shadow areas are present. As shown in the figure, EGMM could detect and remove most of shadows at the relatively small cost of reduced true positives, whereas PAWCS as a top ranked model in Table IV did not give satisfactory results for large shadow removal.

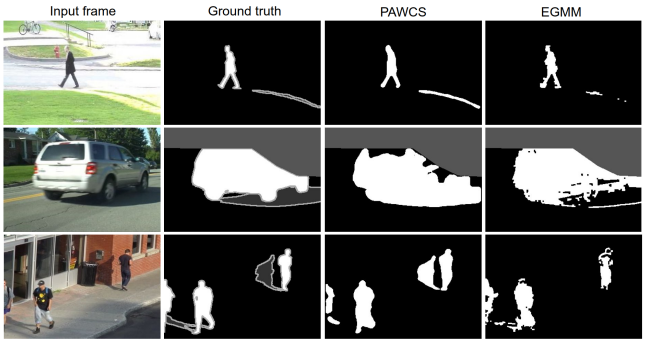


Fig. 5. Sample images of background subtraction results. The first column contains images which have relatively large moving shadows. The second contains ground truth images where dark gray areas indicate shadows. The third and fourth columns show the corresponding results of PAWCS and EGMM methods.

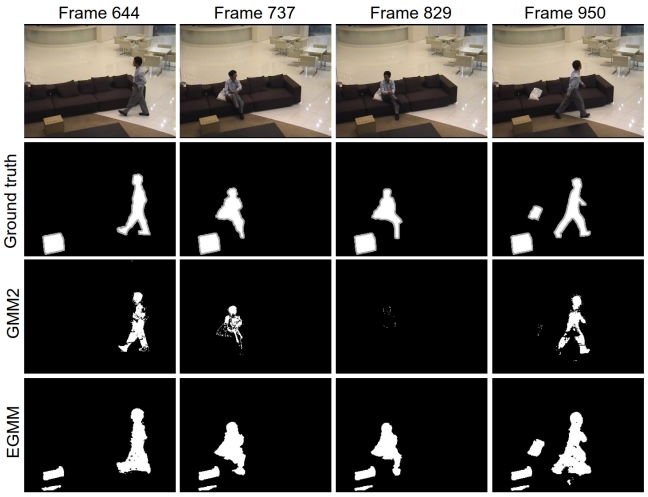


Fig. 6. Comparative results for intermittent objects using the "sofa" video sequence in the 2012 CDnet dataset. The columns correspond to the frame numbers in the video sequence and the rows show the ground truth and the results given by GMM2 and EGMM.

Figure 6 shows the comparative results for intermittent object motion. In this "sofa" video sequence, a person walks with a bag in his hand, sits on a sofa for about 200 frames, and then stands up and walks again. It is clear to see that GMM2 quickly incorporates the person and the bag into the background and hence produced false negative results. By applying different learning rates to the foreground and background regions, EGMM can still detect the person and bag which are stopped temporarily.

## V. CONCLUSIONS

EGMM is an effective background subtraction method especially in illumination change cases and real-time applications requiring HD and higher resolutions.

## REFERENCES


[1] B. Garcia-Garcia, T. Bouwmans, and A. J. R. Silva, "Background subtraction in real applications: Challenges, current models and future directions," *Computer Science Review*, vol. 35, p. 100204, 2020.

[2] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *Proceedings. 1999 IEEE computer society conference on computer vision and pattern recognition (Cat. No PR00149)*, vol. 2. IEEE, 1999, pp. 246–252.

[3] Z. Zivkovic, "Improved adaptive gaussian mixture model for background subtraction," in *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, vol. 2. IEEE, 2004, pp. 28–31.

[4] Z. Chen and T. Ellis, "A self-adaptive gaussian mixture model," *Computer Vision and Image Understanding*, vol. 122, pp. 35–46, 2014.

[5] M. Shah, J. D. Deng, and B. J. Woodford, "Video background modeling: recent approaches, issues and our proposed techniques," *Machine vision and applications*, vol. 25, no. 5, pp. 1105–1119, 2014.

[6] O. Tezcan, P. Ishwar, and J. Konrad, "Bsuv-net: A fully-convolutional neural network for background subtraction of unseen videos," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2020, pp. 2774–2783.

[7] M. Babaee, D. T. Dinh, and G. Rigoll, "A deep convolutional neural network for video sequence background subtraction," *Pattern Recognition*, vol. 76, pp. 635–649, 2018.

[8] L. A. Lim and H. Y. Keles, "Foreground segmentation using convolutional neural networks for multiscale feature encoding," *Pattern Recognition Letters*, vol. 112, pp. 256–262, 2018.

[9] B. Klare and S. Sarkar, "Background subtraction in varying illuminations using an ensemble based on an enlarged feature set," in *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*. IEEE, 2009, pp. 66–73.

[10] P. KaewTraKulPong and R. Bowden, "An improved adaptive background mixture model for real-time tracking with shadow detection," in *Video-based surveillance systems*. Springer, 2002, pp. 135–144.

[11] N. Friedman and S. Russell, "Image segmentation in video sequences: A probabilistic approach," in *Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence*, ser. UAI'97. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1997, p. 175–181.

[12] G.-A. Bilodeau, J.-P. Jodoin, and N. Saunier, "Change detection in feature space using local binary similarity patterns," in *2013 International Conference on Computer and Robot Vision*. IEEE, 2013, pp. 106–112.

[13] P.-L. St-Charles and G.-A. Bilodeau, "Improving background subtraction using local binary similarity patterns," in *IEEE winter conference on applications of computer vision*. IEEE, 2014, pp. 509–515.

[14] B. O. V. O. M. ViBe, "a universal background subtraction algorithm for viedo sequences," *IEEE Transactions on Image Processing*, vol. 20, no. 6, pp. 1709–1724, 2011.

[15] P.-L. St-Charles, G.-A. Bilodeau, and R. Bergevin, "Subsense: A universal change detection method with local adaptive sensitivity," *IEEE Transactions on Image Processing*, vol. 24, no. 1, pp. 359–373, 2014.

[16] ——, "A self-adjusting approach to change detection based on background word consensus," in *2015 IEEE winter conference on applications of computer vision*. IEEE, 2015, pp. 990–997.

[17] S.-h. Lee, G.-c. Lee, J. Yoo, and S. Kwon, "Wisenetmd: Motion detection using dynamic background region analysis," *Symmetry*, vol. 11, no. 5, p. 621, 2019.

[18] M. Braham and M. Van Droogenbroeck, "Deep background subtraction with scene-specific convolutional neural networks," in *2016 international conference on systems, signals and image processing (IWSSIP)*. IEEE, 2016, pp. 1–4.

[19] Y. Wang, Z. Luo, and P.-M. Jodoin, "Interactive deep learning method for segmenting moving objects," *Pattern Recognition Letters*, vol. 96, pp. 66–75, 2017.

[20] D. Sakkos, H. Liu, J. Han, and L. Shao, "End-to-end video background subtraction with 3d convolutional neural networks," *Multimedia Tools and Applications*, vol. 77, no. 17, pp. 23 023–23 041, 2018.

[21] M. C. Bakkay, H. A. Rashwan, H. Salmane, L. Khoudour, D. Puig, and Y. Ruichek, "Bscgan: Deep background subtraction with conditional generative adversarial networks," in *2018 25th IEEE International Conference on Image Processing (ICIP)*. IEEE, 2018, pp. 4018–4022.

[22] J.-Y. Kim and J.-E. Ha, "Foreground objects detection using a fully convolutional network with a background model image and multiple original images," *IEEE Access*, vol. 8, pp. 159 864–159 878, 2020.

[23] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[24] M. O. Tezcan, P. Ishwar, and J. Konrad, "Bsuv-net 2.0: Spatio-temporal data augmentations for video-agnostic supervised background subtraction," *IEEE Access*, vol. 9, pp. 53 849–53 860, 2021.

[25] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001*, vol. 1. Ieee, 2001, pp. I–I.

[26] J. Barreto, P. Menezes, and J. Dias, "Human-robot interaction based on haar-like features and eigenfaces," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, vol. 2. IEEE, 2004, pp. 1888–1893.

[27] S.-H. Huang and S.-H. Lai, "Detecting faces from color video by using paired wavelet features," in *2004 Conference on Computer Vision and Pattern Recognition Workshop*. IEEE, 2004, pp. 64–64.

[28] N. Kanopoulos, N. Vasanthavada, and R. L. Baker, "Design of an image edge detection filter using the sobel operator," *IEEE Journal of solid-state circuits*, vol. 23, no. 2, pp. 358–367, 1988.

[29] A. Ekbal and S. Saha, "Weighted vote-based classifier ensemble for named entity recognition: a genetic algorithm-based approach," *ACM Transactions on Asian Language Information Processing (TALIP)*, vol. 10, no. 2, pp. 1–37, 2011.

[30] C. Zhang and Y. Ma, *Ensemble machine learning: methods and applications*. Springer, 2012.

[31] J. Kittler, M. Hatef, R. P. Duin, and J. Matas, "On combining classifiers," *IEEE transactions on pattern analysis and machine intelligence*, vol. 20, no. 3, pp. 226–239, 1998.

[32] S. Nadimi and B. Bhanu, "Physical models for moving shadow and object detection in video," *IEEE transactions on pattern analysis and machine intelligence*, vol. 26, no. 8, pp. 1079–1087, 2004.

[33] R. Cucchiara, C. Grana, M. Piccardi, and A. Prati, "Detecting moving objects, ghosts, and shadows in video streams," *IEEE transactions on pattern analysis and machine intelligence*, vol. 25, no. 10, pp. 1337–1342, 2003.

[34] R. Cucchiara, C. Grana, M. Piccardi, A. Prati, and S. Sirotti, "Improving shadow suppression in moving object detection with hsv color information," in *ITSC 2001. 2001 IEEE Intelligent Transportation Systems. Proceedings (Cat. No. 01TH8585)*. IEEE, 2001, pp. 334–339.

[35] N. Goyette, P.-M. Jodoin, F. Porikli, J. Konrad, and P. Ishwar, "Changedetection. net: A new change detection benchmark dataset," in *2012 IEEE computer society conference on computer vision and pattern recognition workshops*. IEEE, 2012, pp. 1–8.

[36] V. J. Montero, W.-Y. Jung, and Y.-J. Jeong, "Fast background subtraction with adaptive block learning using expectation value suitable for real-time moving object detection," *Journal of Real-Time Image Processing*, vol. 18, no. 3, pp. 967–981, 2021.

[37] M. Van Droogenbroeck and O. Paquot, "Background subtraction: Experiments and improvements for vibe," in *2012 IEEE computer society conference on computer vision and pattern recognition workshops*. IEEE, 2012, pp. 32–37.

[38] Z. Zivkovic and F. Van Der Heijden, "Efficient adaptive density estimation per image pixel for the task of background subtraction," *Pattern recognition letters*, vol. 27, no. 7, pp. 773–780, 2006.

[39] T. S. Haines and T. Xiang, "Background subtraction with dirichlet-process mixture models," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 4, pp. 670–683, 2013.