BACKGROUND SUBTRACTION FOR STATIC & MOVING CAMERA

Hasan Sajid and Sen-Ching Samson Cheung
hasan.sajid@uky.edu, sccheung@ieee.org

Department of Electrical & Computer Engineering, University of Kentucky, KY, USA

ABSTRACT

Background subtraction is one of the most commonly used components in machine vision systems. Despite the numerous algorithms proposed in the literature and used in practical applications, key challenges remain in designing a single system that can handle diverse environmental conditions. In this paper we present Multiple Background Model based Background Subtraction Algorithm as such a candidate. The algorithm was originally designed for handling sudden illumination changes. The new version has been refined with changes at different steps of the process, specifically in terms of selecting optimal color space, clustering of training images for Background Model Bank and parameter for each channel of color space. This has allowed the algorithm’s applicability to a wide variety of challenges associated with change detection including camera jitter, dynamic background, Intermittent Object Motion, shadows, bad weather, thermal, night videos etc. Comprehensive evaluation demonstrates the superiority of algorithm against state of the art.

Index Terms—Background Subtraction, background modeling, binary classifiers, background model bank

1. INTRODUCTION

Background Subtraction (BS) is one of pre-processing steps in many computer vision applications such as surveillance, tracking, privacy etc. A typical BS process involves building a background model of the scene and then for an input image produce a binary mask to separate the foreground (FG) from the background (BG).

In real world videos, BS is a challenging problem because of variations associated within the scene and camera itself. Challenges associated with scene can be because of dynamic background, low visibility due to bad weather or night time, gradual or sudden illumination changes, intermittent object motion and shadows casted by objects. On the other hand, camera jitter, camera pan-tilt-zoom (PTZ) movement and other noise are challenges posed by camera itself.

Due to these multitude of challenges, there is no single algorithm that can address all of the aforementioned challenges single handedly. Therefore, we present an improved version of our original algorithm in [1] that is able to cope with these challenges. The new version is named as Universal Background Subtraction System version 0 (UBSS V0). For evaluation purposes, we employ a publicly available dataset recently established by the vision community for change detection problem named CDnet 2014 [2]. For fair comparison against the state-of-the-art, the organizers of CDnet recommend a set of common evaluation metrics and have maintained the results of a large set of state-of-the-art algorithms on the common dataset.

The main contribution of this paper is a universal change detection system with innovations in terms of background modeling and use of appropriate color spaces. Unlike other algorithms, which have single model of the background, we construct multiple pixel-wise BG models of the scene but with frame level constraint. This allows the proposed system to account for inter-pixel spatial dependencies and hence extends its applicability to both static and moving camera scenarios.

The use of multiple color spaces is motivated by human visual system that uses two types of cells for color vision under different lighting conditions; rods and cones. Rods play their part under low lighting conditions, whereas cones are responsible during sufficient lighting. During intermediate lighting both of them act together [31]. Analogous to rods and cones, RGB color space plays its part during low lighting conditions since chromatic information is more uniformly distributed across all three channels. YCbCr fails during low lighting conditions (low RGB values) because dark pixels are close to origin and in RGB space chromaticity lines meet at the origin. This increases misclassification chances of dark pixels since color points can be considered close to any chromaticity line [32, 33]. YCbCr is responsible under sufficient lighting and increases FG segmentation accuracy with addition of color channels. Lastly, both of them complement each other during intermediate lighting level.

Another major contribution of the paper is extensive experimentation and evaluation of proposed system against state of the art on CDnet 2014 dataset.

The rest of the paper is organized as follows. Related work is presented in section 2. UBSS V0 is discussed in section 3, followed by experiments and results in section 4. Lastly in section 5 paper is concluded with future directions.

2. RELATED WORK

Background subtraction is a very well-studied topic. There are numerous existing BS algorithms which can be broadly classified into pixel based, region based or frame based category.

The first category of algorithms consists of those relying on pixel-wise statistical model of the scene or background. Some of the methods mentioned in [3, 4] employ simple statistics such as mean and median whereas others use more complex multimodal distributions. The most popular algorithms in this category are pixel-wise Gaussian Mixture Models (GMM) [5] and Kernel Density Estimates (KDE) [6]. It is worth noticing that many extensions and improved versions of [5] have been presented. Examples include [7, 8, 9, 10] and among which, one of the top ranked algorithms at CDnet, Flux Tensor with Split Gaussian (FTSG) [7] method, also uses GMM approach. Pixel based algorithms in general suffer from loss of inter-pixel spatial dependencies and address them by constantly updating model or parameters. Despite the shortcoming of finding appropriate
update rate, this is the most dominant category in background subtraction.

The second category include all the region based methods that capture spatial relationship among pixels. Examples of such methods are [11, 12, 13]. In [12], the authors construct a non-parametric KDE of background and foreground pixels and incorporate pixel location by using posteriori Markov Random Field framework. The key challenge of this method is in determining the appropriate time interval for model update since events can occur at different speeds. [12] and [13] are based on an underlying assumption that neighboring pixels undergo similar variation and therefore considers blocks of different sizes. A difficulty with such methods is to determine the appropriate block size.

The last category are frame based methods. These methods create background model of the scene by considering the entire video frame. Many of these methods are based on a shading model that calculates ratio of intensity between input image and background model often known as quotient image. Examples of such methods are [14, 15, 16]. In [14], the authors model quotient image as a GMM, resulting in a Statistical Illumination (SI) model. The GMM is then incorporated into a probabilistic framework that takes into account color and texture cues.

Another frame based method is Eigen Background (EB) [15]. It builds an eigenspace of background by using training images and then reconstructs BG image by projecting the input image on learned eigenspace. One of potential problems is that it may fail if an input image cannot be represented as a linear combination of training images. Lastly, [16] is primarily combination of EB and SI methods. In comparison to previous two categories, frame based methods are most robust against illumination changes but have been less popular than pixel based category.

### 3. UBSS V0 Description

In this section we briefly provide an overview of UBSS V0 and also detail parameter training and color space selection.

#### 3.1. System Overview

The proposed system has six steps as depicted in Figure 1. Each step is detailed below.

##### 3.1.1. System Initialization

The first step is to use training images to form a Background Model Bank (BMB), train system parameters and select the optimal color space.

A BMB comprises of multiple single mode Gaussian \((\mu_n(X), \sigma_n^2(X))\) models, where \(X\) denote the image coordinates and \(n \in \{1, \ldots, N\}\). Training images are first clustered into \(N\) groups based on correlation measure using K-means algorithm. This is followed by estimation of a single mode Gaussian for each cluster.

In order to train parameters including \(N\) and select the optimal color space, a subset of training images with foreground and their respective ground truths are employed. Details are provided in section 3.2.

##### 3.1.2. Background Model Selection

Next step is to select an appropriate BG model from BMB. For an input image \(I(X)\), BG model is chosen as:

\[
\text{Corr} = \arg \max_{n=1, \ldots, N} \left( \frac{(I - \mu_n)(\mu_n - \mu)' \sqrt{(I - \mu_n)(I - \mu)' \sqrt{\mu_n' - \mu} \mu_n' - \mu}}{(I - \mu_n)(I - \mu)' \sqrt{\mu_n' - \mu} \mu_n' - \mu} \right)
\]

where, \(I\) and \(\mu_n\) are vector forms of \(I(X)\) and \(\mu_n(X)\) respectively. \(\mu_i\) and \(\mu\) are defined as:

\[
\mu_i = \frac{1}{k} \sum_j I_j \quad \text{and} \quad \mu = \frac{1}{k} \sum_j \mu_n j
\]

\(k\) is the total number of pixels of the input frame.

##### 3.1.3. Binary Mask (BM) Generation

Once BG model is chosen, it is passed to Background Subtraction modules along with input image. The modules are also known as Binary Classifiers (BC) and as the name suggests each of the BC produces a binary mask \(D_{\text{mask}}(X)\) for each of the color channels \(D\) of a color space. Further details of background subtraction modules can be found in [1].

##### 3.1.4. Binary Masks Aggregation

All of the BMs produced by BCs are aggregated into a Foreground Detection (FGD) mask as follows:

\[
FGD_{\text{mask}}(X) = \left\lfloor \sum_D D_{\text{mask}}(X) \right\rfloor > t
\]

where \(t\) is a threshold and is different for both RGB and YCbCr color spaces. If both color spaces are used, FGD mask is simply a logical OR of FG mask for each of the color spaces.

##### 3.1.5. Binary Masks Purging

In this step all of the BMs are purged using FGD mask and new BMs \(D_{\text{mask}}(X)\) are obtained:

\[
D_{\text{mask}}(X) = D_{\text{mask}}(X) \cdot \text{Dilate}(FGD_{\text{mask}}(X))
\]

##### 3.1.6. Foreground Mask

This is the final step in which foreground mask is obtained using \(D_{\text{mask}}(X)\):

\[
FG_{\text{mask}}(X) = \left\lfloor \sum_D D_{\text{mask}}(X) \right\rfloor > t
\]

In case both color spaces are used, FG mask is obtained by logical OR of FG mask for each of the color spaces.

![Figure 1. Universal background Subtraction System](image-url)
3.2. Parameter Training and Color Space Selection

In this section, we discuss the use of a subset of training images (typically 3 to 5) with FG and their ground-truths in training the system parameters used in our algorithm as presented in Section 3.1: Number of BG models (N), parameter for pixel level BC (c_{PIdn}), parameter for pixel & frame level BC (c_{FFIdn}), threshold value (t) and also select the optimal color space.

First we build BMBs with N ranging from 2 to 100. Then for each N, we run BCs on the training images and determine the values of c_{PIdn}, c_{FFIdn}, c_{PIdn} and c_{FFIdn}, which result in least error (E).

$$E = \sqrt{(AN - TN)^2 + (AP - TP)^2}$$

where, AN is the Actual Negative or BG pixels in the mask, AP is the Actual Positive or FG pixels in the mask, TN is True Negative and TP stands for True Positive. Since all the three BCs are independent of each other, c_{PIdn}, c_{FFIdn}, c_{PIdn} and c_{FFIdn} are determined independently. Once c_{PIdn}, c_{FFIdn}, c_{PIdn} and c_{FFIdn} are set, we vary threshold value t for both RGB and YCbCr color spaces and choose the one with least E. Finally the set of parameters and color space (RGB, YCbCr or Both) that offers best results in terms of E is chosen.

4. EXPERIMENTS AND RESULTS

In this section, we compare proposed system with 15 state-of-the-art algorithms on publicly available test sequences. We describe the dataset, list parameters and present quantitative and qualitative results.

4.1. CDnet 2014 Dataset

CDnet 2014 is the most comprehensive dataset made available for change detection algorithms. It comprises of 11 challenges: Baseline(BL), Dynamic Background(DB), Camera Jitter(CJ), Intermittent Object Motion(IOM), Shadow(SHD), Thermal(TB), Bad Weather(BW), Low FrameRate(LFR), Night Videos(NV), Pan Tilt Zoom(PTZ) and Turbulence(TB). Each category has 4 to 6 videos totaling to 53 video test sequences. In addition, both training and testing data for each sequence are defined so that it is consistent across all the algorithms employing CDnet 2014 dataset.

The authors of dataset use seven metrics for evaluation purposes: Recall(Re), Specificity(Sp), False Positive Rate(FPR), False Negative Rate(FNR), Percentage of Wrong Classifications(PWC), Precision(Pr) and F-Measure(FM). To compare among different algorithms, these seven metrics are then combined into overall average rank (R) and average rank across categories (RC). For specific description of these metrics refer to [2].

4.2. Parameter Setting

Parameters are determined automatically based on the procedure outlined in section 3.2. Since it is impossible to tabulate all of the parameters for such a large dataset, we discuss the two most meaningful ones: average number of BG models N and the most selected color space for each category.

The number of BG models is less than 10 for most of the categories except categories that involve variations in scene or due to camera such as CJ, PTZ and DB. For CJ, PTZ and DB, average N was found to be 25. The optimal color space for most of the categories is use of both RGB & YCbCr, whereas, as expected for NV, RGB is most favored since YCbCr fails during low lighting conditions. In thermal category, RGB is a natural choice as Cb and Cr channels of YCbCr color space are useless. Lastly, YCbCr is favored most for PTZ category as color information seems to significantly increase foreground segmentation.

4.3. Quantitative Evaluation

This section provides a detailed comparison against state of the art algorithms. Figure 2 depicts segmentation results of UBSS V0 on some of example frames. Complete videos of our proposed system for different categories of CDnet 2014 can be found at our website1. Table 1 provides F-Measure based overall and category-wise comparison. Complete results are presented in Table 2.

We choose to focus on F-Measure for comparing algorithms because of following reasons. First the authors of [2] have observed close correlation between F-measure and ranks of algorithm and in general is accepted as a good indicator for comparison. Second, due to non-linearity of overall ranks, ranks can be affected substantially in case of addition or removal of new methods [17]. Third, ranking is based on two reciprocal metrics; FPR and Sp and therefore would favor ‘precise’ methods [17]. Lastly, since background subtraction is an unbalanced binary classification problem, using PWC metric would result in bias towards more ‘precise’ methods [17].

4.4. Discussions

First, we consider category-wise performance of UBSS V0 against other state of the art. In BL, DB and SHD categories, our algorithm is not among top 3 but produces acceptable results with F-Measure of 92.8%, 79% and 77.8% respectively. According to [17], F-Measure ≥ 80% is considered an acceptable result. In LFR category, F-Measure is 0.627 and marginally less than 0.644 of top performing method but still it is not an acceptable result. The performance of our algorithm is worse in only one of the four test sequences in this category i.e. ‘port_0_17fps’ resulting in F-Measure to drop down. This particular scene is recorded at 0.17 fps with wavering lighting conditions and intense dynamic behavior of water and boats. In NV category, we are 2nd but like other algorithms, the results are affected by low visibility and halos and reflections caused by strong head- lights. In moving camera categories; PTZ and CJ, our algorithm produces best results. It is important to note that in PTZ, we have significant difference in comparison to other state of the art. One of the

---

1 https://www.sites.google.com/site/hasansajid/research#TOC-MB2S-A-Universal-background-Subtraction-System
prime reasons is the frame level approach for background modelling and therefore unlike other state of the art, proposed method does not rely on static camera assumption. Our results are likely to be further improved with the incorporation of a model update mechanism since there are many scenarios that occur in test data which are not a part of training data. Model update is a part of our future work.

In IOM category we are 3rd. This category involves objects being placed and removed at different times and therefore requires model update. Due to lack of model update mechanism we are not able to achieve F-Measure of 80%. With incorporation of such a mechanism results are expected to further improve. Another category i.e. TB also requires model update mechanism. The proposed system performs well in 3 test sequences but we have poor performance in one of the test sequences in which background is changing continuously and hence requires continuous update. Lastly, in TH and BW categories, we are 2nd and 3rd respectively and achieve acceptable results.

Now we analyze overall performance of UBSS V0 against other state of the art. Table 1 clearly depicts that our algorithm is 3rd out of 15 state of the art algorithms. It is important to note that other than top 2 methods none has been able to achieve an overall F-Measure greater than 70%. Please note that all of the statistics and results for UBSS V0 and other state of the art algorithms are available and have been obtained from CDnet 2014 official website³.

Lastly, we consider performance of UBSS V0 with 3 additional configurations in terms of choice of color space i.e. if we use RGB only (UBSS V0-RGB), YCbCr only (UBSS V0-YCbCr) and RGB & YCbCr (UBSS V0-Both). Overall F-Measure of UBSS V0-RGB came out to be 0.627, 0.628 for UBSS V0-YCbCr and lastly 0.671 for UBSS V0-Both. The results clearly indicate higher overall F-Measure (0.713) when appropriate color space is selected.

5. CONCLUSION AND FUTUREWORK

We have presented a universal change detection system that comprises of background model bank and computationally inexpensive binary classifiers. The use of optimal color space for scene in question has not only increased foreground segmentation accuracy but has made it robust across different change detection challenges. Comprehensive evaluation indicates UBSS V0 as being one of top performing methods on CDnet 2014 benchmark. The frame level approach for background modeling makes our algorithm superior against state of the art for moving camera categories.

Current version of UBSS V0 lacks model update mechanism, which is a part of future work. The incorporation of such mechanism is expected to further increase segmentation accuracy. Currently the system is implemented in MATLAB and in most coarse form offers on average 8.5 frames per second for images of 320x240 resolution on a core i5 PC with 8GB RAM. The simplicity of our algorithm, optimization of code and implementation in C++ is expected to result in real time capability.

---

Tables:

**Table 1. Overall and Category-wise comparison on CDnet 2014 Dataset²**

<table>
<thead>
<tr>
<th>Category</th>
<th>Average Recall</th>
<th>Average Specificity</th>
<th>Average FPR</th>
<th>Average FNR</th>
<th>Average PWC</th>
<th>Average Precision</th>
<th>Average F-Measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bad Weather</td>
<td>0.8211</td>
<td>0.9936</td>
<td>0.0064</td>
<td>0.1789</td>
<td>0.9449</td>
<td>0.7571</td>
<td>0.7730</td>
</tr>
<tr>
<td>Low FrameRate</td>
<td>0.6656</td>
<td>0.9942</td>
<td>0.0058</td>
<td>0.3344</td>
<td>0.8610</td>
<td>0.6604</td>
<td>0.6279</td>
</tr>
<tr>
<td>Night Videos</td>
<td>0.5535</td>
<td>0.9773</td>
<td>0.0227</td>
<td>0.4465</td>
<td>0.3692</td>
<td>0.4899</td>
<td>0.5158</td>
</tr>
<tr>
<td>PTZ</td>
<td>0.5770</td>
<td>0.9945</td>
<td>0.0055</td>
<td>0.4230</td>
<td>0.7821</td>
<td>0.4988</td>
<td>0.5118</td>
</tr>
<tr>
<td>Turbulence</td>
<td>0.5571</td>
<td>0.9981</td>
<td>0.0019</td>
<td>0.4429</td>
<td>0.3086</td>
<td>0.6804</td>
<td>0.5698</td>
</tr>
<tr>
<td>Baseline</td>
<td>0.9158</td>
<td>0.9979</td>
<td>0.0021</td>
<td>0.0842</td>
<td>0.4361</td>
<td>0.9431</td>
<td>0.9287</td>
</tr>
<tr>
<td>Dynamic Background</td>
<td>0.7637</td>
<td>0.9972</td>
<td>0.0028</td>
<td>0.2363</td>
<td>0.4848</td>
<td>0.8060</td>
<td>0.7694</td>
</tr>
<tr>
<td>Camera Jitter</td>
<td>0.8321</td>
<td>0.9929</td>
<td>0.0071</td>
<td>0.1679</td>
<td>1.5408</td>
<td>0.8443</td>
<td>0.8367</td>
</tr>
<tr>
<td>Intermittent Object Motion</td>
<td>0.6386</td>
<td>0.9931</td>
<td>0.0069</td>
<td>0.3614</td>
<td>3.1858</td>
<td>0.8201</td>
<td>0.7092</td>
</tr>
<tr>
<td>Shadow</td>
<td>0.7762</td>
<td>0.9918</td>
<td>0.0082</td>
<td>0.2238</td>
<td>1.5794</td>
<td>0.8063</td>
<td>0.7784</td>
</tr>
<tr>
<td>Thermal</td>
<td>0.8101</td>
<td>0.9908</td>
<td>0.0092</td>
<td>0.1899</td>
<td>1.5315</td>
<td>0.8174</td>
<td>0.8115</td>
</tr>
<tr>
<td>Overall</td>
<td>0.7192</td>
<td>0.9929</td>
<td>0.0071</td>
<td>0.2808</td>
<td>1.3922</td>
<td>0.7435</td>
<td>0.7139</td>
</tr>
</tbody>
</table>

**Table 2. Complete results of UBSS V0 on CDnet 2014 Dataset²**

² In each column, Red font is for best, Green Font for second best and Blue font represents third best result.

³ www.changedetection.net
6. REFERENCES


