Abstract

Background modeling in fast changing scenarios is a challenging task due to unexpected events like sudden illumination changes, reflections, and shadows, which can strongly affect the accuracy of the foreground detection. In this paper, we describe a real-time and effective background modeling approach, called FAFEX, that can deal with global and rapid changes in the scene background. The method is designed to identify variations in the background geometry of the monitored scene and it has been quantitatively tested on a publicly available data set, containing a varied set of highly dynamic environments. The experimental evaluation demonstrates how our method is able to effectively deals with challenging sequences in real-time.

1. Introduction

Background subtraction (BS) is widely used as base layer for a number of Computer Vision applications, spacing from surveillance and monitoring of dynamic scenes to augmented reality. BS methods have to be robust with respect to the multiple aspects of the problem, including illumination changes (e.g., clouds), shadows, camera jitter (e.g., due to winds), background movement (e.g., waves on the water surface), and permanent or temporary changes in the background geometry (e.g., moved furniture in a room) and event recognition.

Many BS techniques can tackle several different issues like shadows, sudden and gradual illumination changes, fast varying and highly dynamic background. However, some issues are still only partially solved due to their inherent complexity. For instance, unexpected illumination changes (e.g., switching on/off room lights or opening a door in a corridor) can cause false positives until the background (BG) model is updated. Furthermore, deciding whether and when the BG model may absorb variations in the background geometry (e.g., a car in a parking lot) affects the results of higher level layers, such as object tracking.

In addition to the large literature (see [3] for a recent survey), a number of BS software libraries have been released. With such a variety of methods, often accompanied by open source implementations, the possibility of carrying out quantitative comparisons is a key aspect. In order to evaluate the accuracy of different BS methods, many data sets have been proposed (e.g., Change Detection\(^1\), Visor\(^2\)) and specific challenges have been presented as well (e.g., Background Models Challenge). Moreover, the chance of generating an accurate foreground (FG) mask represents a fundamental requirement for higher level tasks, such as object tracking and event recognition.

The main contributions of this paper are: 1) the description of an online clustering BS method called Fast Adaptive Foreground Extraction (FAFEX) and 2) its evaluation on a challenging and publicly available data set. FAFEX is specifically designed to deal with nonregular and high frequency noise, dynamic background. Unexpected events involving the entire frame (e.g., due to rain or clouds) are handled by considering the time behavior of the pixels in the observed scene [4]. To this end, we adopt a solution based on a temporal filtering of the background model, that allows to deal with global changes in the scene.

The remainder of this paper is structured as follows. Related work is described in Section 2. The proposed method is presented in Section 3 and the details of the model update mechanism are given in Section 4. The experimental evaluation is shown in Section 5, while conclusions are drawn in Section 6.

2. Related Work

A fundamental requirement to deal with highly dynamic background is to integrate a pixel-wise statistical model approach in combination with a global model
of the movement in the scene [1]. The background model has to be sensitive enough to detect the moving objects, while adapting to long-term lighting variations (e.g., time of the day) and structural changes (e.g., objects entering the scene and becoming stationary). The model needs also to rapidly adjust in case of sudden background changes, like light switching. Combining local and global models allows to satisfy simultaneously both the sensitivity to foreground motion and the ability to model sudden background changes. From the large literature on BS algorithms, we decided to discuss some methods adopting frame level solutions that are robust to unexpected events. Furthermore, since our algorithm is clustering-based, we also describe some methods that exploit the same concept.

Frame-level Solutions. A method for addressing the problem of global changes in the scene has been proposed by Shah et al. [9]. The authors describe an illumination invariant background model using Gaussian Mixture Models (GMM), in order to adapt local parameters for stationary objects, and a hierarchical SURF feature matching, for dealing with ghosts and illumination changes. Yoshinaga et al. [13] apply a GMM for extracting statistical illumination invariant local features, in order to adapt to illumination as well as dynamic background changes. Vosters et al. [12] propose a real-time approach, which combines Eigenbackground with a statistical illumination model. The method is based on two algorithms: the first one is used to reconstruct the background frame, while the second one improves the foreground segmentation. However, the moved background objects will always be detected as foreground after movements. This is because the object’s new location is not incorporated into the Eigenspace background model.

Clustering Approaches. Fan et al. [6] perform a k-means clustering and single Gaussian model to reconstruct the background through a sequence of scene images with foreground objects. Then, based on the statistical characteristics of the background pixel regions, the algorithm detects the moving objects. In addition, an adaptive algorithm for foreground detection is used in combination with morphological operators and a region-labeling mechanism. Kumar and Sureshkumar [8] propose a modified k-means algorithm to compute background subtraction in real-time. In their experimental results, the algorithm shows that selecting centroids can lead to a better background subtraction, being efficient and robust for dynamic environment having new objects. Moreover, in comparison with traditional methods in real world, the experiments show that the approach is more accurate than other classical algorithms.

Differently to the above mentioned clustering methods, time is a key factor in FAFEX. Indeed, the background model is built by considering multiple frame samples that are collected on the basis of a time period and of a persistence map. Each pixel is classified as foreground or background depending on a persistence interval, thus allowing to deal with changes in the background geometry.

3. Fast Adaptive Background Modeling

The main idea behind FAFEX is the incremental computation of the BG model through the discretization of the color distribution for each pixel, by using an on-line clustering approach. The functional architecture of the proposed modeling is described in Fig. 1. The BG model is incrementally generated by analyzing a variable size sample set of image frames. The size of the sample and the duration of the sampling period is controlled by the BG Model Manager module. At steady state, a new BG model is generated every milliseconds, where is the number of sample frames and is a time period in ms. Fig. 2 shows the increment background modeling mechanism, that is carried out before reaching the steady state. A first background model is built by using frames only, where . Next, a second model is generated after milliseconds, where . A new model is incrementally computed by increasing the sampling period until reaching the final value , after which the model can be considered as stable.

Algorithm 1 describes the details of the proposed background modeling phase. For each pixel , is a set of pairs , where is the discrete
representation of a set of adjacent colors in a given color space (e.g., a range in RGB or HSV space) and \( f(c) \) is the number of occurrences of \( c \) (i.e., of colors within the range represented by \( c \)) in the sample set. After processing all the sample set, only those colors that have enough occurrences are maintained in the background model.

In this way, the BG model contains, for each pixel, a discrete and compact multi-modal representation of the color probability distribution of the background, which does not require to fit the data in some predefined distribution (e.g., Gaussian). This is the main difference with respect to a Mixture of Gaussians approach, where fitting Gaussian distributions is required and typically the number of Gaussians is determined a priori and limited [10, 14].

Once the BG model \( \mathcal{B} \) is computed, the foreground mask \( F \) is determined by using a parallelized thresholding method. Indeed, the background modeling is performed independently for each pixel \( p(i, j) \), allowing to decompose and distribute the foreground computation of the current image. The parameters passed to the BG model generation algorithm are:

- The association threshold \( a \). A pixel \( p \) in position \( (i, j) \) is considered a foreground point if the condition \( \text{dist}(c, \mathcal{B}(i, j)) \geq a \) holds;
- The number \( n_0 \) of scene samples to be analyzed;
- The minimal number \( d \) of occurrences to consider a color value \( c \) to be a significant background value.

The Validation module is responsible for the computation of the filtered foreground mask \( \tilde{F} \), that is obtained by filtering out possible false positives due to shadows and reflections in the scene, as described in [2].

The BG Model Manager is responsible for the detection of fast changes in the image. If the percentage of foreground pixels with respect to the total number of pixels in the image is above a given threshold \( \gamma \), then a plausible sudden change is occurring in the scene. We experimentally found that 80% is a good value for \( \gamma \) and that varying the value for \( \gamma \) in the range [60%, 90%] does not affect substantially the foreground extraction process. The Model Update process is described in the next section.

4. Model Update

Fast changes affecting large regions of the scene can happen during the creation of a new BG model. As an example, Fig. 3 shows the BMC sequence 422, in which the presence of fog makes the current BG model completely unusable (see the foreground mask generated by the frame 507). Moreover, in such a situation, the model currently being computed becomes useless, since it models a past scenario that is now changed.

We adopt two solutions for coping with modifications in the background, due to changes in the background geometry, like a parked car, or to sudden global variations, such as light switching:

1. **Conditional Update** handles variations in the background geometry by labeling pixels as potential foreground points;
2. **Temporal Filtering** for quickly adapting to sudden global changes.

**Conditional Update.** Elgammal et al. [5] proposed two alternative strategies to update the background. In **selective update**, only pixels classified as belonging to the background are updated, while in **blind update** every pixel in the background model is updated. The selective (or conditional) update improves the detection of the targets since foreground information are not added to the BG model, thus solving the problem of ghost observations. However, when using selective updating any
incorrect pixel classification produces a persistent error, since the BG model will never adapt to it. Blind update does not suffer from this problem since no update decisions are taken, but it has the disadvantage that the values not belonging to the background are added to the model.

We propose a different solution, aiming at solving the problems of both selective and blind update. Given a scene sample $S_k$ and the current FG mask $M$, if $M(i,j) = 1$ and $S_k(i,j)$ is associated to a background mode in the BG model under development, then it is labeled as a “foreground mode”. When computing the foreground, if $p(i,j)$ is associated with a foreground mode, then $p$ is classified as a potential foreground point. Such a solution allows for identifying regions of the scene representing not moving foreground objects, as the gray pixels depicted in Fig. 3.

The decision about including or not the potential foreground points as part of the background is taken on the basis of a persistence map. If a pixel is classified as potential foreground consecutively for a period of time longer than a predefined value $T$ (e.g., $T = 10$ seconds), then it becomes part of the BG model. An example of selective update is shown in Fig. 4, where the background modified by a parked car leaving the scene is correctly modeled by FAFEX, thus avoiding ghost detections. The same idea can be used to deal with modifications of the background due to foreground objects that become stationary, as in the case of a car entering the scene and parking. It can be considered as part of the background if it does not move for a (sufficient) period of time $T$. Until the period $T$ has not passed, the points belonging to that foreground object may be considered as “potential foreground points”.

Furthermore, the labeling process provides additional information to higher level modules (e.g., a visual tracking module) helping in reducing ghost observations.

Temporal Filtering. Fig. 2 describes the situation in which a background change occurs at time $t$, during the computation of a new model. The change is detected

![Figure 4. Example of ghost observation on the BMC video 001: A parked car leaves the scene. In the first column are reported the frames 1447 and 1560, in the second one the FG masks generated by the MOG2 [14] function with ghost detections, and in the third one the output produced by FAFEX.](image)

![Figure 5. Example of temporal filtering on the BMC sequence 422. (a) Frame 456. (b) Basic FAFEX without temporal filtering. (c) Advanced FAFEX with temporal filtering.](image)

![Figure 6. Three example frames from the synthetic image sequences. (a) Cloudy street. (b) Sunny rotary. (c) Foggy street.](image)

![Figure 5 shows an example of comparison between the output generated by FAFEX without (Fig. 5b) and with (Fig. 5c) the temporal filtering feature.](image)

5. Experimental Results

A quantitative evaluation of the proposed method has been conducted by using the publicly available Background Models Challenge (BMC) data set\(^3\). The BMC data set contains 20 synthetic and 9 real video sequences, annotated with ground-truth data. The synthetic video sequences (see Fig. 6) include two differ-

\(^3\)bmc.iut-auvergne.com
ent urban scenarios (i.e., a street and a rotary). For each scene, a particular event (e.g., sun uprising or fog) is inserted in the sequence, in order to represent a fast changing background. The set of real videos is made of 9 challenging sequences acquired from static cameras in common video-surveillance operational scenarios including a parking area, a train station, and a warehouse (Fig. 7).

The source code of the FAFEX method is available at www.dis.uniroma1.it/~bloisi/software/fafex.zip.

5.1. Metrics

A set of widely used metrics have been considered in order to represent different kinds of quality of a BS method [11].

**F-measure** is a static quality metric. The results are analyzed in terms of: false negatives (FN), false positives (FP), true positives (TP), and true negatives (TN). It is computed as:

\[
F\text{-}measure = \frac{1}{n} \sum_{i=1}^{n} \frac{Prec_i \times Rec_i}{Prec_i + Rec_i}
\]

where \(i\) represents the current frame and:

\[
\begin{align*}
Prec_i &= (1/2)(PrecP_i + PrecN_i) \\
Rec_i &= (1/2)(RecP_i + RecN_i) \\
PrecN_i &= TN_i / (TN_i + FN_i) \\
PrecP_i &= TP_i / (TP_i + FP_i) \\
RecN_i &= TN_i / (TN_i + FN_i) \\
RecP_i &= TP_i / (TP_i + FP_i)
\end{align*}
\]

**Peak Signal-Noise Ratio (PSNR)** is another static quality metric. Given a ground-truth video sequence \(G\), a set \(S\) representing the \(n\) images computed by means of a BS algorithm, and a frame \(i\), PSNR is defined as:

\[
PSNR = \frac{1}{n} \sum_{i=1}^{n} 10 \log_{10} \frac{m}{\sum_{j=1}^{m} \|S_i(j) - G_i(j)\|^2}
\]

where \(S_i(j)\) is the \(j\)-th of image \(i\) (having size \(m\)) in the sequence \(S\) (having length \(n\)).

**Structural SIMilarity (SSIM)** can be used to consider the output of the BS process in a more perceptual way. Let \(\mu_S\) and \(\mu_G\) be the means, \(\sigma_S\) and \(\sigma_G\) be the standard deviations, and \(\text{cov}_S,G\) be covariance of \(S\) and \(G\). SSIM can be computed as:

\[
SSIM(S,G) = \frac{1}{n} \sum_{i=1}^{n} \frac{(2\mu_S\mu_G + c_1)(2\text{cov}_S,G + c_2)}{(\mu_S^2 + \mu_G^2 + c_1)(\sigma_S^2 + \sigma_G^2 + c_2)}
\]

where \(c_1 = (k_1 \times L)^2\), \(c_2 = (k_2 \times L)^2\), \(L = 255\), \(k_1 = 0.01\), and \(k_2 = 0.03\).

**D-score** is a measure that is sensitive to medium range errors. These errors have a great impact on the object recognition phase, because can lead to modifications on object’s shape. Values close to zero for the **D-Score** measure demonstrate a good performance of the tested algorithm. It is computed as:

\[
D\text{-}score(S_i(j)) = \exp \left[ -\log_2 \left( 2DT(S_i(j)) - \frac{5}{2} \right)^2 \right]
\]

where \(DT(S_i(j))\) is a function for obtaining the minimal distance between the pixel \(S_i(j)\) and the nearest reference point.

5.2. Quantitative Evaluation

In order to provide a quantitative measure of the capacity of FAFEX to deal with challenging sequences, we report in Table 1 a comparison of FAFEX with other methods. In particular, four method are considered for the comparison: MOG [7] and MOG2 [14] from the OpenCV library, and the best two performing methods presented at the BMC Workshop during the ACCV conference in 2012, namely Yoshinaga [13] and Shah [9]. We tested FAFEX by using all the sequences of the BMC data set, and in this paper, we focus on the 10 sequences that contain sudden change conditions (see Table 1).

The results are obtained with the same configuration parameters for all sequences and they show that FAFEX has comparable detection rates with respect to the other considered methods. In particular, for the evaluation sequences, FAFEX is the best performing methods in almost all the scenarios. It is worth noting that the other two methods by Yoshinaga and Shah do not have real-time performance on the BMC data set.

**Computability.** In order to ensure real-time performance, we measured the computational speed of FAFEX on all the sequences used for the experiments by using all 8 cores of an Intel(R) Core(TM) i7-3610QM CPU @ 2.30GHz, 8 GB RAM. The results are reported in Table 2. FAFEX is a multi-core algorithm developed in C++ by exploiting the OpenMP library\(^4\).

\(^4\)http://openmp.org/wp/
Table 1. Results of the quantitative comparison on the BMC data set.

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Method</th>
<th>Recall</th>
<th>Precision</th>
<th>F-measure</th>
<th>PSNR</th>
<th>D-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>122</td>
<td>MOG</td>
<td>0.59</td>
<td>0.95</td>
<td>0.77</td>
<td>48.80</td>
<td>0.29</td>
</tr>
<tr>
<td></td>
<td>MOG2</td>
<td>0.88</td>
<td>0.93</td>
<td>0.90</td>
<td>48.82</td>
<td>0.18</td>
</tr>
<tr>
<td></td>
<td>Yoshinaga</td>
<td>0.86</td>
<td>0.89</td>
<td>0.88</td>
<td>53.01</td>
<td>0.09</td>
</tr>
<tr>
<td></td>
<td>Shah</td>
<td>0.93</td>
<td>0.90</td>
<td>0.91</td>
<td>48.20</td>
<td>0.17</td>
</tr>
<tr>
<td></td>
<td>FAFEX</td>
<td>0.90</td>
<td>0.92</td>
<td>0.92</td>
<td>49.02</td>
<td>0.13</td>
</tr>
<tr>
<td>212</td>
<td>MOG</td>
<td>0.63</td>
<td>0.85</td>
<td>0.73</td>
<td>54.04</td>
<td>0.18</td>
</tr>
<tr>
<td></td>
<td>MOG2</td>
<td>0.85</td>
<td>0.96</td>
<td>0.90</td>
<td>59.68</td>
<td>0.09</td>
</tr>
<tr>
<td></td>
<td>Yoshinaga</td>
<td>0.83</td>
<td>0.88</td>
<td>0.85</td>
<td>52.76</td>
<td>0.10</td>
</tr>
<tr>
<td></td>
<td>Shah</td>
<td>0.92</td>
<td>0.89</td>
<td>0.91</td>
<td>52.71</td>
<td>0.10</td>
</tr>
<tr>
<td></td>
<td>FAFEX</td>
<td>0.97</td>
<td>0.98</td>
<td>0.92</td>
<td>53.90</td>
<td>0.07</td>
</tr>
<tr>
<td>222</td>
<td>MOG</td>
<td>0.59</td>
<td>0.96</td>
<td>0.73</td>
<td>49.80</td>
<td>0.29</td>
</tr>
<tr>
<td></td>
<td>MOG2</td>
<td>0.84</td>
<td>0.95</td>
<td>0.89</td>
<td>54.19</td>
<td>0.18</td>
</tr>
<tr>
<td></td>
<td>Yoshinaga</td>
<td>0.82</td>
<td>0.77</td>
<td>0.80</td>
<td>50.34</td>
<td>0.09</td>
</tr>
<tr>
<td></td>
<td>Shah</td>
<td>0.93</td>
<td>0.91</td>
<td>0.92</td>
<td>48.69</td>
<td>0.15</td>
</tr>
<tr>
<td></td>
<td>FAFEX</td>
<td>0.91</td>
<td>0.94</td>
<td>0.93</td>
<td>49.36</td>
<td>0.12</td>
</tr>
<tr>
<td>312</td>
<td>MOG</td>
<td>0.62</td>
<td>0.93</td>
<td>0.74</td>
<td>54.66</td>
<td>0.17</td>
</tr>
<tr>
<td></td>
<td>MOG2</td>
<td>0.78</td>
<td>0.62</td>
<td>0.69</td>
<td>44.59</td>
<td>0.20</td>
</tr>
<tr>
<td></td>
<td>Yoshinaga</td>
<td>0.85</td>
<td>0.64</td>
<td>0.73</td>
<td>44.30</td>
<td>0.12</td>
</tr>
<tr>
<td></td>
<td>Shah</td>
<td>0.87</td>
<td>0.91</td>
<td>0.89</td>
<td>53.13</td>
<td>0.11</td>
</tr>
<tr>
<td></td>
<td>FAFEX</td>
<td>0.93</td>
<td>0.93</td>
<td>0.88</td>
<td>53.63</td>
<td>0.09</td>
</tr>
<tr>
<td>422</td>
<td>MOG</td>
<td>0.56</td>
<td>0.51</td>
<td>0.54</td>
<td>30.84</td>
<td>0.43</td>
</tr>
<tr>
<td></td>
<td>MOG2</td>
<td>0.72</td>
<td>0.56</td>
<td>0.63</td>
<td>35.44</td>
<td>0.45</td>
</tr>
<tr>
<td></td>
<td>Yoshinaga</td>
<td>0.88</td>
<td>0.88</td>
<td>0.89</td>
<td>48.33</td>
<td>0.18</td>
</tr>
<tr>
<td></td>
<td>Shah</td>
<td>0.80</td>
<td>0.75</td>
<td>0.77</td>
<td>45.32</td>
<td>0.17</td>
</tr>
<tr>
<td></td>
<td>FAFEX</td>
<td>0.89</td>
<td>0.92</td>
<td>0.80</td>
<td>47.26</td>
<td>0.19</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Method</th>
<th>Recall</th>
<th>Precision</th>
<th>F-measure</th>
<th>PSNR</th>
<th>D-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>001</td>
<td>MOG</td>
<td>0.56</td>
<td>0.53</td>
<td>0.55</td>
<td>35.43</td>
<td>0.35</td>
</tr>
<tr>
<td></td>
<td>MOG2</td>
<td>0.66</td>
<td>0.62</td>
<td>0.64</td>
<td>37.87</td>
<td>0.94</td>
</tr>
<tr>
<td></td>
<td>Yoshinaga</td>
<td>0.93</td>
<td>0.82</td>
<td>0.87</td>
<td>47.94</td>
<td>0.98</td>
</tr>
<tr>
<td></td>
<td>Shah</td>
<td>0.71</td>
<td>0.70</td>
<td>0.70</td>
<td>35.61</td>
<td>0.99</td>
</tr>
<tr>
<td></td>
<td>FAFEX</td>
<td>0.94</td>
<td>0.91</td>
<td>0.89</td>
<td>35.20</td>
<td>0.09</td>
</tr>
</tbody>
</table>

6. Conclusions

We presented a real-time adaptive BG method, called FAFEX, based on a temporal filtering process. This filtering phase allows to handle global and sudden changes in the background, as well as unexpected events that change the background in a significant way. A quantitative experimental evaluation, carried out on a publicly available data set of challenging video sequences, demonstrates how FAFEX can generate accurate FG masks in real-time, even in presence of highly dynamic scenarios. Finally, we compared quantitatively our algorithm by using four different metrics with respect to two OpenCV functions and the best two performing methods benchmarked in the BMC 2012 competition, obtaining comparable results, but with a significantly lower computational cost.

References