Global Illumination Compensation for Background Subtraction
Using Gaussian-based Background Difference Modeling

Julien A. Vijverberg\textsuperscript{1,2}, Marijn J.H. Loomans\textsuperscript{1,2}, Cornelis J. Koeleman\textsuperscript{1}, Peter H.N. de With\textsuperscript{2,3}

\{julien,marijn,rick\}@vdg-security.com, p.h.n.de.with@tue.nl

\textsuperscript{1}VDG Security B.V., Zoetermeer, The Netherlands
\textsuperscript{2}Eindhoven University of Technology, Eindhoven, The Netherlands
\textsuperscript{3}CycloMedia Technology B.V., Waardenburg, The Netherlands

Abstract

This paper presents a background segmentation technique, which is able to process acceptable segmentation masks under fast global illumination changes. The histogram of the frame-based background difference is modeled with multiple kernels. The model that represents the histogram at best, is used to determine the shift in luminance due to global illumination or diaphragm changes, such that the background difference can be compensated. Experimental results have revealed that the number of incorrectly classified pixels using global illumination compensation instead of only the approximated median method reduces from 77\% to 19\% shortly after a fast change. The performance of the proposed technique is similar to state-of-the-art related work for global illumination changes, despite the fact that only luminance information is used. The algorithm is computationally simple and can operate at 30 frames-per-second for VGA resolution on a P-IV 3-GHz PC.

1. Introduction

The continuing trend to increase the computation power of processors, allows a new generation of security cameras to be equipped with powerful Digital Signal Processors (DSPs) and/or reconfigurable logic like Field Programmable Gate Arrays (FPGAs). These DPSs and FPGAs enable security cameras to improve the compression of high-resolution video and allows content analysis algorithms to detect relevant events.

In video-content analysis such as applied within surveillance systems, background segmentation is commonly used to extract the relevant objects from an input image. Important requirements for background segmentation include that (1) the number of pixels correctly classified should be high, (2) the background should be updated regularly, while relevant objects should not distort the background and (3) the computational complexity should be low. Examples where traditional background segmentation techniques tend to fail are background objects that are changed or removed, vegetation moved by the wind, variations in weather, illumination changes and stationary objects.

Since stationary and slowly moving objects tend to appear in the background image, the quality of the segmentation mask degrades. A simple counter measure is to decrease the update rate of the background estimation algorithm. However, when the background changes quickly (often due to global illumination changes), the slowly-updated background image looses its validity for the current situation. Therefore, we need to model and compensate the global illumination changes in the sequence.

In literature on background segmentation, Stauffer and Grimson \cite{13} and Elgammal \textit{et al.} \cite{3} were the first in using advanced density estimation techniques on pixel-level; Gaussian Mixture Model (GMM) and non-parametric techniques respectively. No global difference models were used. In \cite{14}, Toyama \textit{et al.} present the Wallflower background estimation, which estimates multiple backgrounds (two in their experiments) and switches between the learned backgrounds when a large number of pixels are detected as foreground, for example due to different illumination conditions. This is a very pragmatic approach, but it does only provide a solution for two states, which also have to be trained in advance.

This has led others \cite{5,7} to follow a more structural approach. In these papers, gradient information was used to classify foreground pixels in two categories: ‘illumination change’ and ‘object present’. Gradient information is most discriminating at the boundaries of the objects, but does not provide a clear difference for large, untextured objects (e.g. a white bus on the road surface). Furthermore, gradient in-
formation used for pixel-based background subtraction is highly unreliable under camera motion (shaking). Even more advanced features like histograms of gradients [10], which are supposed to be robust against small camera movements, appeared to be very sensitive to vibrations of the camera in our experiments using i-LIDS parking vehicle sequences [6]. Seki et al. [12] model the co-occurrence of variations between adjacent blocks, thereby achieving accurate segmentation under illumination variation and swaying vegetation, but this proposal seems computationally intensive.

Pilet et al. [11] took an approach similar to ours to handle sudden global illumination changes. They compute the ratio of the current image and background image for each color channel and compute two other texture-based features. Consecutively, they model the PDF of these features as a mixture of Gaussians using an expectation-maximization algorithm, resulting in 18 fps for $360 \times 288$ pixels and 5-6 fps for $752 \times 576$ pixels on a 2-GHz CPU. These results are considered to be unsatisfactory low frame rates.

In this paper, we present a technique to compensate the background difference for global illumination and diaphragm changes. The proposed technique fits a number of models on the background difference and selects the most probable model. The parameters of the selected model are used to compute the shift due to global illumination and diaphragm changes. Improving on earlier methods, the proposed method has a good performance-to-complexity ratio working only on gray-level images, it does not rely on gradient information thereby making the illumination estimation more robust to camera vibrations, and it does not require the new global illumination state to be learned in advance, provided that the new state fits one of the models.

The next section contains a description of the foreground segmentation method, in which the proposed global illumination compensation operates. Section 3 describes the global illumination models and the estimation/compensation technique. In Section 4, the proposed method is compared to the segmentation results without global illumination compensation and state-of-the-art background estimation techniques. Finally, we present our conclusions.

2. Segmentation Pipeline

To understand the context of the proposed method, it is necessary to describe the other parts of the segmentation pipeline. A block diagram of the segmentation algorithm is shown in Fig. 1. Our segmentation algorithm is based on the usage of motion vectors and a simple background estimation and subtraction algorithm. Our objective is to execute this application on a camera with a powerful DSP in parallel with a video compression engine. We reuse the motion vectors provided by the video compression engine.

For our experiments, the 3DRS [2] motion estimator has been selected.

First, the motion vectors are compensated for global motion (GMC). The compensated motion vectors are used to generate an initial segmentation mask (MASK). This initial segmentation mask provides robustness to withstand background movement.

The second part of the pipeline estimates the background $B$ (in the BGEST block) using the initial segmentation mask and then subtracts the current image $I$ from the background $B$, where the result is thresholded to obtain another segmentation mask. This mask can be used in addition to the initial segmentation mask to provide better segmentation results for slowly moving and stationary objects. In our experiments, we use the Approximate Median (AM) [9] for background estimation.

The update period of the AM algorithm is set to 64 frames in all sequences. As explained in Section 1, there is a trade-off between the update rate (and the quality of the segmentation mask during global illumination changes) and the ability to segment stationary objects. We find it sufficient to say that it is evident that the performance of the AM will increase during global illumination changes, if the update rate is increased.

The proposed technique has been been marked with the dotted ellipse in Fig. 1. Our technique adds background difference histogram block (BGHIST) and illumination estimation block (ILLU.EST) and it slightly modifies the background subtraction. The details are described in the next section.

3. Illumination Estimation and Compensation

For each pixel on position $(x, y)$, we first compute the background difference $\Delta(x, y) = I(x, y) - B(x, y)$. Consecutively, the histogram of the background difference $h_\Delta[u]$ is computed for all pixels with $u \in [-255, 255]$. Hence, the number of bins $N_\pi = 511$. During initial experiments, we observed that the shape of the histogram $h_\Delta$ was often similar to a Gaussian distribution, a mixture of two Gaussian distributions or a Laplacian distribution. For
this reason, we have decided to fit these three models on the
histogram and select the best matching model.

In the sequel, a description is given of how the three models are fit onto the histogram \( h_\Delta \) (Section 3.1), the selection of the best model (Section 3.2) and how the background difference is compensated to obtain a segmentation mask (Section 3.3).

### 3.1. Illumination Parameter Estimation

First, the parameters \( \theta = \{w, \mu, \sigma\} \) for a Laplacian distribution

\[
\hat{h}_\Delta[u; \theta] = we^{-|u-\mu|/\sigma},
\]

and a Gaussian-shaped function

\[
\hat{h}_\Delta[u; \theta] = w e^{-(u-\mu)^2 / 2\sigma^2},
\]

are computed. For the both histogram models, the \( \mu \) parameter is set to the median of the measured histogram. For the Laplacian distribution, \( \sigma \) is set to \( \sum_u |h_\Delta[u] - \mu| / N_s \) and \( w \) is set to \( \frac{1}{2\sigma} \sum u h_\Delta[u] \). For the Gaussian distribution, we have initialized \( \sigma \) with the square root of the variance, and \( w \) with \( \frac{1}{\sqrt{2\pi}\sigma} \sum u h_\Delta[u] \).

Second, we estimate the parameters of a two-component GMM. A two-component GMM better covers the diaphragm changes and complicated difference histograms. We apply the Levenberg-Marquardt (LM) algorithm to estimate the parameters and use the implementation from [8]. This optimization algorithm iteratively finds the minimum Sum of Squared Distances (SSD) using the Jacobian of the SSD. For our experiments, the parameters of the components are initialized, using the parameters of the single Gaussian function \( \theta_{1,2} = \{\frac{1}{2}, \mu \pm 10, \sigma / \sqrt{2}\} \).

### 3.2. Illumination Model Selection

Once the parameters of the models are estimated, the SSD is computed for each of them. As the SSD is a type of error metric, it is directly clear that the one-component model with the largest SSD should not be selected. However, since the two-component model is likely to better fit the histogram \( h_\Delta \), we need a method to select either the one-component or two-component model, while penalizing the two-component model for the additional parameters. Both the Bayesian Information Criterion (BIC) and Akaike’s Information Criterion (AIC) provide a method to penalize additional parameters [4]. Because experiments showed no large differences in performance using the BIC or AIC, for our experiments we have adopted the AIC specified by

\[
\text{AIC} = N_s \log_c (\text{SSD} / N_s) + N_p,
\]

where \( N_s = 255 \) is the number of bins in the histogram and \( N_p \in \{3, 6\} \) the number of parameters of the model. According to [4] many authors applied the AIC to non-linear problems. For this reason, we have also applied Eq. (3) to our non-linear optimization problem.

### 3.3. Illumination Compensation

The illumination compensation is performed by computing the thresholds on the background difference to select between foreground \( (255) \) and background \( (0) \). Suppose that a model was selected with \( K \) mixture components and that model \( k \) (for \( k = 0, \ldots, K - 1 \)) has mean \( \mu_k \).

\[
F(x, y) = \begin{cases} 
255 & \text{if } |\Delta(x, y) - \mu_k| > T_k \forall k \\
0 & \text{otherwise}
\end{cases}
\]

The threshold \( T_k \) is set to \( \max(T_{\min}, 1.5\sigma_k) \), such that the allowed difference increases with the variance but cannot be smaller than \( T_{\min} = 16 \). Despite the fact that the additional thresholds increase the complexity of the subtraction step compared to the basic algorithm \( |f(x, y) - B(x, y)| \), the overall complexity is still low.

### 4. Experiments

The maximum number of iterations of the LM optimization is set to 10. Fig. 2 shows that after 10 iterations, the number of correctly classified pixels is almost constant. The peak at 2 and 3 iterations can be explained by the fact that the desk area in the the bottom-right part of the image (see Fig. 4(a)) does not fit the illumination model, but is correctly classified as background during the iterations 2 and 3. In other sequences, we also observed no significant differences between 10 or 100 iterations, but we could not quantify this, due to a lack of ground-truth data.

The output of the \texttt{BGSUB} block (Fig. 1) is shown in Fig. 3(a), 4(a) and 5(a) for the ‘AVSS PV Medium’, the ‘Wallflower’ light switch and a custom sequence, respectively. Each of these figures shows, from left to right, the input frame, the segmentation mask using illumination compensation and the segmentation mask without illumination compensation.

![Figure 2](image.png)
In addition, we compare the results to the segmentation mask generated by Multi-Modal Mean (MMM) \cite{1} which is a state-of-the-art background estimation technique intended for embedded vision systems. In the experiments, we use 4 modes and vary the update period between 1 frame, 16 frames, and 64 frames. In the remainder of this paper, these will be abbreviated by MMM-1, MMM-16 and MMM-64, respectively. Figures 2(b-d), 3(b-d), 4(b-d) show results obtained with the MMM technique.

4.1. Accuracy

Fig. 3 shows a still of the ‘AVSS PV Medium’ sequence \cite{6}. This video sequence was recorded with a vibrating camera and contains multiple stationary objects during global illumination changes. Fig. 3(b)-(d) show that under these circumstances, both MMM and the proposed technique provide segmentation masks of acceptable quality.

Fig. 4 shows the results on the light-switch sequence, which is one of the ‘Wallflower’ sequences \cite{14}. These
results show that the proposed technique is better than the uncompensated difference, but even the segmentation mask created using compensation is not very good. However, our results are similar to the results by Pilet et al. [11]. This is partly due to the flickering computer screen and the chair being pulled away such that new background is revealed. To compare the results with the ground truth mask, the results are down-sampled to $160 \times 120$ pixels. Table 1 shows the number of incorrectly classified pixels.

It has to be noted that the use of a single ground-truth frame has a large impact on the segmentation results. For all update periods, the MMM algorithm adapts the background after the light switch. However, during the frames before the ground-truth frame, MMM-16 adapts slightly faster than the other two. In addition, after MMM has adapted to the new situation (lights turned on/off), it handles the flickering computer screen much better than the AM method and the proposed technique.

Since only some of the sequences available exhibit global illumination changes, we have also captured custom sequences to test the proposed algorithm. Fig. 5 shows the results for our custom sequence, which contains many global illumination changes due to weather conditions and diaphragm changes. Fig. 5(b)-5(d) shows the corresponding segmentation results of the MMM method.

For test sequences not shown in this paper exhibiting less illumination changes, the proposed technique did not noticeably degrade the quality of the segmentation mask. This phenomenon was also noticed for the algorithm experimented by Pilet et al. [11]. However, the proposed technique fails on occasions (notably on some parts of the

<table>
<thead>
<tr>
<th>method</th>
<th>percentage incorrect</th>
</tr>
</thead>
<tbody>
<tr>
<td>MMM-1</td>
<td>60%</td>
</tr>
<tr>
<td>MMM-16</td>
<td>19%</td>
</tr>
<tr>
<td>MMM-64</td>
<td>56%</td>
</tr>
<tr>
<td>w/o compensation</td>
<td>77%</td>
</tr>
<tr>
<td>with compensation</td>
<td>19%</td>
</tr>
<tr>
<td>Pilet et al.</td>
<td>23%*</td>
</tr>
</tbody>
</table>

Table 1. Percentage of incorrectly classified pixels for the Wallflower light-switch sequence at frame 1865. The percentage generated for Pilet et al. is an estimation by the authors based on the numbers in the paper by Pilet et al. [11].

‘AVSS PV Medium’ sequence), where the model has not been estimated correctly. This usually happens when the algorithm selects the single Gaussian option, when two Gaussians close to each other would be more appropriate. 1

4.2. Computational Complexity

The algorithm is executed on a single core of a P-IV 2.4-GHz quad-core PC. Table 2 shows the average execution time per frame for each sequence. The execution time includes the time for 3DRS and the complete algorithm as depicted in Fig. 1. The column headed by Time(100) contains the average execution time for a run of upto 100 iterations of the LM algorithm and the column headed by Time(10) for a run of upto 10 iterations. The achieved frame rate for

Example sequences are available at http://vca.ele.tue.nl/demos/avss2009/index.html
Table 2. Average execution time (ms) to process a single frame.

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Width</th>
<th>Height</th>
<th>Time (100)</th>
<th>Time (10)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVSS PV Medium</td>
<td>752</td>
<td>576</td>
<td>103</td>
<td>40</td>
</tr>
<tr>
<td>Light Switch</td>
<td>640</td>
<td>480</td>
<td>43</td>
<td>32</td>
</tr>
<tr>
<td>Custom</td>
<td>512</td>
<td>384</td>
<td>86</td>
<td>16</td>
</tr>
</tbody>
</table>

We distinguish between a maximum of 100 and 10 iterations of the LM optimization procedure to show the relation between the number of iterations and the execution time.

10 iterations varies between 25 and 62 fps. The large differences between the execution time for 100 iterations or 10 iterations imply that the execution time is highly dependent on the number of iterations of the LM algorithm. This illustrates the complexity of the LM algorithm and that our technique can be executed in real time, when the number of iterations is limited.

5. Conclusion

We have presented a technique for more robust background segmentation handling global illumination changes. This technique is based on estimating the PDF of the background difference using two Gaussian models and one Laplacian model. The algorithm selects one of these models on a frame basis, using the AIC and the SSD metric.

In all cases, using illumination compensation yields improvements over the segmentation mask produced by the plain segmentation algorithm. In addition, we have compared the proposed segmentation technique against the MMM background estimation technique. This comparison has shown that MMM can provide a good segmentation mask, if the update rate is tuned to the situation. However, using MMM, stationary objects require special attention, as they may be omitted occasionally. The experiments showed that the proposed technique and MMM are comparable in performance. MMM outperforms our technique in cases of flickering screens and moving vegetation, whereas our technique outperforms MMM during fast global illumination changes. Both techniques appeared to be not very sensitive to camera vibration in our experiments, in contrast to techniques using gradient information.

We have also compared the proposed segmentation technique against the global illumination compensation technique proposed by Pilet et al. [11] and found comparable performance. Significant differences are the ability to handle shadows (Pilet et al. are able to ignore shadows) and the computational complexity: 6 fps for Pilet et al. vs. 21 fps for the proposed algorithm on e.g. a 2.0-GHz PC. Finally, we add that Pilet et al. require the use of gradient information, which we consider disadvantageous, as it is sensitive to camera vibrations. Besides this aspect, our technique is more attractive for surveillance cameras with embedded video analysis, since our algorithmic complexity is much lower and we favour the robustness on illumination changes.

References