A Likelihood-Based Background Model for Real Time Processing of Color Filter Array Videos

Vito Renó(✉), Roberto Marani, Nicola Mosca, Massimiliano Nitti, Tiziana D'Orazio, and Ettore Stella

Institute of Intelligent Systems for Automation,
Italian National Research Council, Bari, Italy
reno@ba.issia.cnr.it

Abstract. One of the first tasks executed by a vision system made of fixed cameras is the background (BG) subtraction and a particularly challenging context for real time applications is the athletic one because of illumination changes, moving objects and cluttered scenes. The aim of this work is to extract a BG model based on statistical likelihood able to process color filter array (CFA) images taking into account the intrinsic variance of each gray level of the sensor, named Likelihood Bayer Background (LBB). The BG model should be not so computationally complex while highly responsive to extract a robust foreground. Moreover, the mathematical operations used in the formulation should be parallelizable, working on image patches, and computationally efficient, exploiting the dynamics of a pixel within its integer range. Both simulations and experiments on real video sequences demonstrate that this BG model approach shows great performances and robustness during the real time processing of scenes extracted from a soccer match.

1 Introduction

Artificial vision systems (AVSs) equipped with fixed cameras usually need to implement the BG subtraction as the first low level computational task. The output of such process generally is the input for a large amount of software modules that can implement, for example, object tracking or scene understanding. Today, the amount of data processed by an AVS can be dramatically huge because state of the art cameras can achieve very high throughputs in the order of Gb/s. Researchers and engineers are investigating on how to move low level computational load directly on smart cameras [2] reducing the amount of information that needs to be transferred on computers for processing purposes.

Generally speaking, BG models can be classified as *Temporal difference methods* and *Background subtraction methods*: the former group obtains the foreground subtracting and thresholding two consecutive frames; the latter group builds a dynamic model that is updated over time and subtracted to each frame that needs to be processed. One of the most used BG subtraction method is the Adaptive Mixture of Gaussians (MoG) proposed by Stauffer and Grimson [13] that uses Gaussian distributions to represent the variation of pixel intensity.

V. Murino et al. (Eds.): ICIAP 2015 Workshops, LNCS 9281, pp. 218–225, 2015.
DOI: 10.1007/978-3-319-23222-5_27

Two examples of subsequent improvements of this algorithm are the MoGv2 [16] that adaptively updates the parameters of the model over time and its variant on Bayer pattern inputs [14]. Other BG algorithms known in literature include: the Eigenbackground [8] introduced by Oliver et al., that models the BG in a vector subspace obtained via PCA; the Codebook [7] proposed by Kim et al., that implements a quantization of the pixel values using codebooks in order to compress the model size; the GMG [5] by Godbehere et al., that estimates the entire pixel intensity distribution rather than its parameters using dynamic information and updating only the probability distributions associated with background pixels; models based on Hidden Markov Models (HMMs) [11] to represent pixel intensity variations as discrete states.

The possibility of implementing code directly on smart cameras opens new research trends applied to AVSs. In this context, a BG model able to work with CFA images can be implemented directly on embedded chips. Computationally efficient and parallelizable operations are required to find the best trade-off between complexity and reliable results in real time. The Adaptive light-weight algorithm detailed in [1] and applied to process atheltic videos is an example of relevant research interest. According to [15], this type of scenes can be used to detect salient events (i.e. offsides or goals during football matches [3,4,6]), analyse and track objects (i.e. ball and players), perform 3D reconstructions or analyse tactics. Therefore, a robust BG needs to be responsive to light changes and fast in the updates, even if it is modelled with a few bootstrapping frames.

In this paper a BG model able to deal with CFA raw images taking into account the intrinsic sensor variance of each gray value is presented. The variance raises as the gray level increases, therefore LBB exploits this information instead of the classical approaches that evaluate single pixels over time. Finally, each Bayer patch is labelled as BG by means of a likelihood-based approach. The rest of the work is organized as follows: in the second section the proposed algorithm is detailed, the third section contains experiments and results carried out on athletic videos and the last one discusses the conclusions and future works.

2 Methodology

2.1 Algorithm Description

LBB is divided in three main building blocks that are summarized in List. 1.1, namely initialization, processing and update. The first step is executed only once and initializes the BG image setting each pixel to half intensity. This all gray logic is due to the absence of any *a priori* knowledge about the scene. The processing phase is composed of: variance, likelihood, fine tuning and energy. The last one is the same presented in [10], while the other are detailed singularly in the following sub sections. The BG image is updated according to PIIB logic [10] enriched by a binary update mask M. Hence, each BG pixel value is increased or decreased by κ if the corresponding M value is set to true (in our implementation $\kappa = 1$). In addition, LBB calculates a second version of the background that does not take care of M (BG_{nu}) with the aim of avoiding ghosts on the scene, as it will be described later.

Listing 1.1. Algorithm pseudocode

```
Background Initialization
for each frame
    Variance process
    for each patch
        Likelihood process
    if(Background is learned)
        Fine tuning process
    Background Update
    Energy Process
```

2.2 Variance Process

The variance considered in the this method is not related to the observations of a single pixel over time, but is a function of the gray level and so it models the different responses of the sensor to different light intensities. Therefore, for each frame, the location of the occurrences of each generic gray value γ is first stored in a set

$$\text{Obs}(\gamma) = \{k = (u, v) | BG(u, v) = \gamma\} \tag{1}$$

Then, the variance V at the time t, associated to the γ-th gray level is iteratively updated with the following formula:

$$V_t(\gamma) = \frac{V_{t-1}(\gamma) \cdot N_{t-1}(\gamma) + \sum_k |I_t(k) - BG(k)|^2}{N_t(\gamma)} \tag{2}$$

where $k \in \text{Obs}(\gamma)$, $N(\gamma)$ is the number of times the γ-th gray level occurred over time and BG is the background. In the equations BG is substituted with the latest available frame (I_{t-1}) while the BG is being learned, namely until the energy gradient descent reaches its minimum value. Fig. 1 shows an example of convergence of this model while estimating μ and σ values of known normal distributions, that will be discussed in the next section.

2.3 Likelihood Process

This task is executed for each Bayer squared patch $P_i = (p_1, \ldots, p_4)^T$ of the image, so that P_i contains two green level values, a red one and a blue one. Considering the pixels as normal independent random variables, the likelihood of observing a background patch given a set of parameters $\theta = (\mu_1, \ldots, \mu_4, \sigma_1, \ldots, \sigma_4)$ can be calculated with the formula:

$$\mathcal{L}(\theta | P_i) = \prod_{j=1}^{4} f_{\mu_j, \sigma_j}(p_j) = \ell_i \tag{3}$$

where $\mu_j = BG(p_j)$, $\sigma_j = V_t(BG(p_j))^{\frac{1}{2}}$ and $f_{\mu_j,\sigma_j}(p_j)$ is the normal probability density function with mean μ_j and standard deviation σ_j computed in p_j. Therefore, the mean value of a pixel is its corresponding BG value, while the variance depends on its gray level, since different intensity values might have different variances. Following the same steps described in the previous section, the BG is substituted with the latest captured frame until the model is in the learning phase. Formally, a threshold $\tau_L = 0$ is used to classify each patch as background or foreground. In our implementation $\tau_L = 10^{-10}$, considering that 0 can not be achieved due to noise and floating point representation issues—experiments show that the value is small enough to guarantee a stable and reliable BG. The binary update mask of a BG patch is set to true, while it is false in case of a foreground patch. This selective update is useful to achieve robustness and to avoid updating when an object is moving on the scene.

2.4 Fine Tuning Process

The fine tuning task takes place only when the BG has been learned by the system and enriches the pipeline with two modules: a cosine similarity filter [9] and a ghost filter. The first one exploits the dot product between two vectors, specifically a foreground Bayer patch (P_f) and its corresponding background (P_b), both $\in \mathbb{N}^4$. The cosine of the angle between the two patches is filtered to blacken the foreground if it is similar to the background according to the following equation:

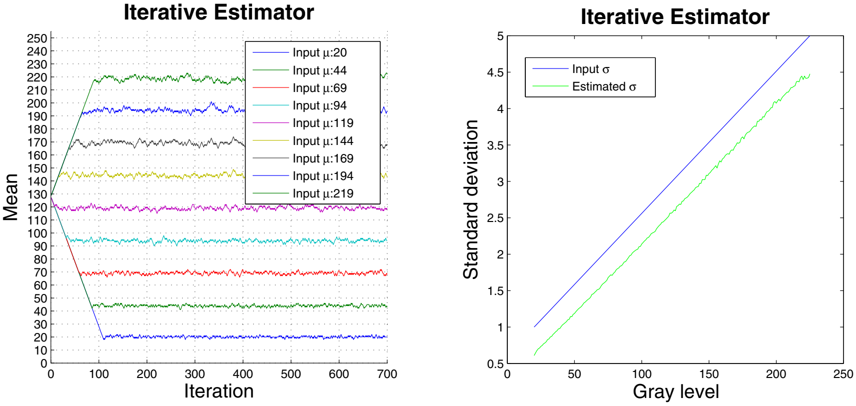
$$P_f = 0 \text{ if } \frac{P_f \cdot P_b}{|P_f||P_b|} > \tau_{\mathrm{S}} \tag{4}$$

where $\tau_{\mathrm{S}} \sim 1$.

The ghost filter is needed when there are no stable background frames at the beginning of a video, i.e. when the bootstrap phase contains almost stationary objects that are likely to be inserted in the background. In these cases, a movement of the object when the BG has been learned causes the presence of a ghost in the foreground. This phenomenon is removed comparing the incoming frame I_t with the background fully updated at each iteration BG_{nu} in correspondence of the ghost patch P_g. If $|I_t(P_g) - BG_{nu}(P_g)| = 0$, then the background is updated setting $BG(P_g) = BG_{nu}(P_g)$.

3 Experiments and Results

The model presented in Sect. 2 has been first tested in Matlab in order to numerically confirm its correctness. For this reason, samples from ~ 200 normal distributions with known (μ, σ) have been extracted. Fig. 1(a) shows that, starting from 128 (half intensity for 1 byte unsigned variables), each estimated mean tends to the input one in ~ 100 frames in the worst case. The distribution with input mean $\mu = 119$ (magenta) converges immediately in a couple of iterations, while for $\mu = 20$ (blue) more iterations are needed to achieve the result. This is due to the update process that consists of unary increments or decrements



(a) μ estimation

(b) σ estimation

Fig. 1. Examples of convergence of the Iterative Estimator of Mean and Standard deviation.

at each iteration, as pointed out in Sect. 2.1. Fig. 1(b) shows the convergence of the standard deviation estimator after $\sim$ 10M iterations. In particular, the estimated σ tends to the input one subtracted by a bias due to the iterative formulation showed in (2). Moreover, LBB has been evaluated against the GMG and MoGv2 algorithms implemented in the BGS Library [12]. The test has been conducted on the same dataset presented in [10], that contains five videos that represent a football match. AR- scenes are focused on the penalty area and the size of each frame is 1600 × 736, while a larger area of 1920 × 1080 pixels is captured in the FG- ones. The five scenes contain some typical situations of a soccer match, for example: a cluttered scene with illumination changes (AR1); the shoot of a penalty kick that implies almost all players around the penalty area (AR2); the shoot of a free kick (AR3) and two actions that are filmed from a wider point of view (FG1 and FG2). In FG2 some players are warming up, so the scene is more dynamic than the one in FG1. Each BG model is evaluated after 20, 40, 60 and 80 seconds after the starting frame f_0.

Fig. 2 contains the qualitative analysis of some frames in terms of ground truth and foreground masks. Rows 1 and 3 contain a cluttered scene where a high number of players is moving in in the penalty area. Here, the MoGv2 (Fig. 2 (d)) shows a weak output due to the constant update of model parameters that is including the players in the BG, while the other approaches (Fig. 2 (c) - (e)) produce a more stable output. The frames in the second row show a scene where the advertising is changing. Here, LBB is updating the BG mask while the other algorithms already did it, due to the updating speed implemented by the unary increment described in Sect. 2.1. This corresponds to the LBB outlier in Fig. 3(a) with low precision and high recall, that gradually tends to a stable configuration when the BG mask is updated. The quantitative results have been extracted calculating the F-Measure, Precision and Recall on four different frames, representing the considered time interval. Each ground truth frame has been manually obtained starting from the raw video. Let TP be the number of true positives pixels, FP be the number of false positives pixels, TN be the number of true negatives pixels and FN be the number of false negatives pixels on the foreground mask. Accordingly, Precision P, Recall R and F-Measure F are defined as:

$$P = \frac{TP}{TP + FP}, \; R = \frac{TP}{TP + FN}, \; F = 2 \cdot \frac{P \cdot R}{P + R} \tag{5}$$

Fig. 3 summarizes the metrics calculated for each video sequence. The comparison of LBB Precision and Recall values against the best value among GMG and MoGv2 shows that the average LBB R value is 39% better than the others, while the P value is generally comparable. This result is shown in Fig. 3(a) where each point in the P-R plane is referred to a run of a specific algorithm (red for MoGv2, blue for GMG and green for LBB). According to this representation, the ground truth has coordinates $(1, 1)$, therefore points in the upper right part of the figure correspond to the best results. High R values for LBB demonstrate that the approach is robust to false negative outputs. Fig. 3(b) shows the 3D bar chart representation of the F-measure. The AR sequences represent a complex situation with a high number of moving people in foreground and here the F-Measure of LBB is higher than the other methods used for the comparison. In the FG ones LBB and GMG behave in a similar way and show comparable results in terms of F-Measure. In particular, the FG1 sequence starts with an advertising change and here LBB has a low F-Measure in the first frame (FG1 - 1) because the foreground mask is noisy, but then the F-Measure increases, so the model is correctly updated in the subsequent frames. The overall average of the LBB F-Measure is 18% better than GMG and MoGv2, thus confirming that the proposed method is capable of modelling such scenarios.

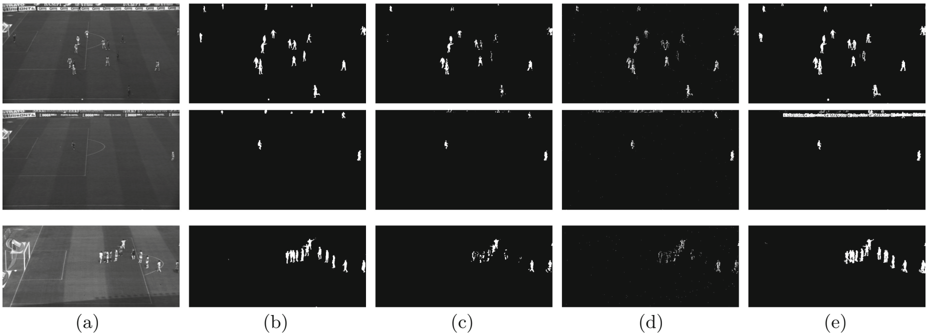
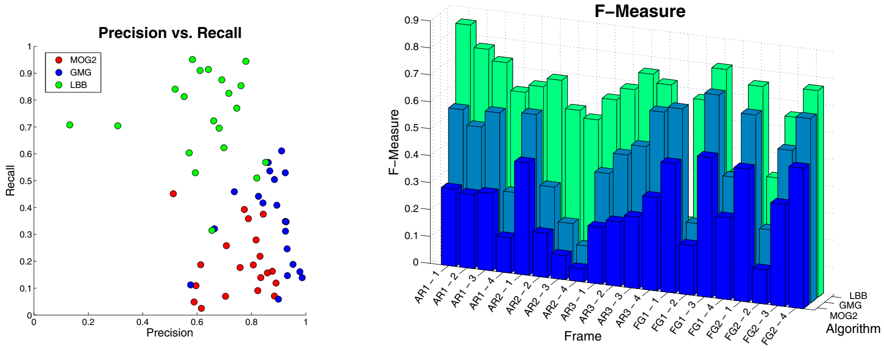


Fig. 2. Qualitative results for some frames. The columns contain, respectively, the original frame (a), the ground truth (b) and the foreground masks obtained with GMG (c), MoGv2 (d) and LBB (e).



(a) (b)

Fig. 3. Quantitative results on the dataset in terms of Precision, Recall 3(a) and F-Measure 3(b).

4 Conclusion

In this paper a likelihood-based background model for real time processing of CFA images is presented. The algorithm is designed with respect to the state of the art output format for vision cameras and implements a statistical model that takes into account the BG as the mean image while modelling the variance of each gray level processing its occurrences in the whole frames. For this reason, the variance is not calculated with respect to the observations of a single pixel over time, but is related to the intrinsic nature of the sensor. Looking at Fig. 3, LBB is able to obtain good performances while processing soccer videos. Even if the precision is lower than the other methods experimented, the recall value is significantly higher, as it is noticeable looking at the position of LBB markers in the P-R plane. These results confirm the robustness of the proposed approach in the athletic video processing context. Finally, the formulation of the algorithm enables its implementation directly on smart cameras (e.g. on FPGA or ARM cpus) and future works will regard both embedding and experimentation on other type of raw videos, for example in the field of surveillance.

References

1. Casares, M., Velipasalar, S., Pinto, A.: Light-weight salient foreground detection for embedded smart cameras. Computer Vision and Image Understanding **114**(11), 1223–1237 (2010). special issue on Embedded Vision
2. Cherian, S., Singh, C., Manikandan, M.: Implementation of real time moving object detection using background subtraction in fpga. In: International Conference on Communications and Signal Processing, ICCSP 2014, pp. 867–871 (2014)
3. D'Orazio, T., Leo, M., Spagnolo, P., Mazzeo, P., Mosca, N., Nitti, M., Distante, A.: An investigation into the feasibility of real-time soccer offside detection from

a multiple camera system. IEEE Transactions on Circuits and Systems for Video Technology **19**(12), 1804–1818 (2009)

4. D'Orazio, T., Leo, M., Spagnolo, P., Nitti, M., Mosca, N., Distante, A.: A visual system for real time detection of goal events during soccer matches. Computer Vision and Image Understanding **113**(5), 622–632 (2009). computer Vision Based Analysis in Sport Environments
5. Godbehere, A., Matsukawa, A., Goldberg, K.: Visual tracking of human visitors under variable-lighting conditions for a responsive audio art installation. In: American Control Conference, ACC 2012, pp. 4305–4312 (2012)
6. Hamid, R., Kumar, R., Hodgins, J., Essa, I.: A visualization framework for team sports captured using multiple static cameras. Computer Vision and Image Understanding **118**, 171–183 (2014)
7. Kim, K., Chalidabhongse, T.H., Harwood, D., Davis, L.: Real-time foreground-background segmentation using codebook model. Real-time Imaging **11**(3), 172–185 (2005)
8. Oliver, N., Rosario, B., Pentland, A.: A bayesian computer vision system for modeling human interactions. IEEE Transactions on Pattern Analysis and Machine Intelligence **22**(8), 831–843 (2000)
9. Qian, G., Sural, S., Gu, Y., Pramanik, S.: Similarity between euclidean and cosine angle distance for nearest neighbor queries. In: Proceedings of the 2004 ACM Symposium on Applied Computing, SAC 2004, pp. 1232–1237. ACM, New York (2004)
10. Renò, V., Marani, R., D'Orazio, T., Stella, E., Nitti, M.: An adaptive parallel background model for high-throughput video applications and smart cameras embedding. In: Proceedings of the International Conference on Distributed Smart Cameras, ICDSC 2014, pp. 30:1–30:6. ACM, New York (2014)
11. Rittscher, J., Kato, J., Joga, S., Blake, A.: A probabilistic background model for tracking. In: Vernon, D. (ed.) ECCV 2000. LNCS, vol. 1843, pp. 336–350. Springer, Heidelberg (2000)
12. Sobral, A.: BGSLibrary: An opencv c++ background subtraction library. In: IX Workshop de Visão Computacional (WVC 2013). Rio de Janeiro, Brazil, June 2013
13. Stauffer, C., Grimson, W.: Learning patterns of activity using real-time tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence **22**(8), 747–757 (2000)
14. Suhr, J.K., Jung, H.G., Li, G., Kim, J.: Mixture of gaussians-based background subtraction for bayer-pattern image sequences. IEEE Transactions on Circuits and Systems for Video Technology **21**(3), 365–370 (2011)
15. Yu, X., Farin, D.: Current and emerging topics in sports video processing. In: IEEE International Conference on Multimedia and Expo, ICME 2005, pp. 526–529 (2005)
16. Zivkovic, Z.: Improved adaptive gaussian mixture model for background subtraction. In: Proceedings of the 17th International Conference on Pattern Recognition, ICPR 2004, vol. 2, pp. 28–31, August 2004