The Importance of Features for Background Modeling and Foreground Detection

Jyoti

M.C. A. dept. M.D. University Rohtak

Abstract: **Background modeling has emerged as a popular foreground detection technique for various applications in video surveillance. Background modeling methods have become increasing efficient in robustly modeling the background and hence detecting moving objects in any visual scene. Although several background subtraction and foreground detection have been proposed recently, no traditional algorithm today still seem to be able to simultaneously address all the key challenges of illumination variation, dynamic camera motion, cluttered background and occlusion. This limitation can be attributed to the lack of systematic investigation concerning the role and importance of features within background modeling and foreground detection. With the availability of a rather large set of invariant features, the challenge is in determining the best combination of features that would improve accuracy and robustness in detection. The purpose of this study is to initiate a rigorous and comprehensive survey of features used within background modeling and foreground detection. Further, this paper presents a systematic experimental and statistical analysis of techniques that provide valuable insight on the trends in background modeling and use it to draw meaningful recommendations for practitioners. In this paper, a preliminary review of the key characteristics of features based on the types and sizes is provided in addition to investigating their intrinsic spectral, spatial and temporal properties. Furthermore, improvements using statistical and fuzzy tools are examined and techniques based on multiple features are benchmarked against reliability and selection criterion. Finally, a description of the different resources available such as datasets and codes is provided.**

Keywords: **Background modeling, Foreground detection Features, Local binary patterns**

I. INTRODUCTION

Background modeling and foreground detection are important steps for video processing applications in video-surveillance [1], optical motion capture [2], multimedia [3], teleconferencing and human–computer interface. The aim is to separate the moving objects, called ‘‘foreground’’, from the static information, called ‘‘background’’. For example, Fig. 1 shows an original frame of a sequence from the BMC 2012 dataset [4], the reconstructed background image and the moving objects mask obtained from a decomposition into the low-rank matrix and sparse matrix based model [5]. Conventional background modeling methods exploit the temporal variation of each pixel to model the background and hence use it in conjunction with change detection for foreground extraction. The last decade witnessed very significant contributions to this field [5–14]. Despite these works and advances to background modeling and foreground detection, the dynamic nature of visual scenes attributed by changing illumination conditions, occlusion, background clutter and noise have challenged the robustness of such techniques. Under this pretext, focus has shifted towards the investigation of features and their role in improving both the accuracy and robustness of background modeling and foreground detection. Although fundamental low-level features such as color, edge, texture, motion and stereo have reported reasonable success, recent visual applications using mobile devices and internet videos where the background is non-static, require more complex representations to guarantee robust moving object detection [15]. Furthermore, in order to generalize existing background modeling and foreground detection schemes to real-life scenes where dynamic variations are inevitable and the pose of the camera is little known, automatic feature selection, model selection and adaptation for such schemes are often desired.

Considering the needs and challenges aforementioned, in this paper, a comprehensive review of low-level and hand-crafted features used in background modeling and foreground detection is initiated for benchmarking them against the complexities of typical dynamic scenes. Thus, the aim of this survey is then to provide a first complete overview of the role and the importance of features in background modeling and foreground detection by reviewing both existing and new ideas for (1) novices who could be students or engineers beginning in the field of computer vision, (2) experts as we put forward the recent advances that need to be improved, and (3) reviewers to evaluate papers in journals, conferences, and workshops. In addition, this survey gives a complete overview Moreover, an accompanying website called the Features Website1 is provided. It allows the reader to have a quick access to the main resources, and codes in the field. So, this survey is intended to be a reference for researchers and developers in industries, as well as graduate students, interested in robust background modeling and foreground detection in challenging environments. A review regarding feature concepts: A first complete overview of low-level and hand-crafted features used in background modeling and foreground detection over the last decade concerning more than 600 papers. After a preliminary overview on the key concepts in the field of features in Section 2, a survey of spectral features including color features are detailed in Section 4. Then, spatial features such as edge, texture and stereo features are studied in Section 5, Section 6 and Section 7, respectively. Temporal features such as motion features are reviewed in Section 8. In Section 15, features that are extracted in alternative domains other than the pixel domain are described.

II. CLASSIFICATION BY SIZE

The size of the picture element chosen for interpreting necessary features that faithfully represent its characteristics plays a crucial role in modeling. As mentioned earlier, features can be computed from and for a pixel [20], a block [21] or a cluster [22]. That is, the size of the picture element that is used to model the background and hence for comparing the current image frame to the background model, can either be a pixel [20], a block [21], a region (Regions of difference [23], shape [24], behavior [25], clus- ter [22], super-pixel [26], global appearance [27]) with a feature value. During practical implementations, a feature value at a given pixel can either depend on the feature value at the pixel itself or on the feature values around a predefined neighborhood in the form of a block or a cluster.

Pixel-based Features: These features, otherwise known as point features, concern only the pixel at a given location (x, y). This is the case of intensity and color features but in some cases include stereo features too. The background model applied in this case of pixel-based modeling and com- parison is an independent process on each individual pixel. Practically, these features are used in uni-modal or multi- modal pixel-wise background modeling and foreground de- tection. Furthermore, pixel-based feature can be used to compute the mean or an other statistic value over spatial and/or temporal neighborhood to take into account spa- tial and/or temporal constraints. Then, the statistic value is assigned to the central pixel. For example, Varadarajan et al. [28,29] proposed a region-based Mixture of Gaussians called (R-MOG) instead of a pixel based MOG. Each region is a square neighborhood which is effectively a block of size 44. Then, the color mean obtained from the neighborhood is assigned to the central pixel. Block-based Features: This category of features is a gen- eralization of the pixel-type, where in the element size a block of 1 1 or any arbitrary block size m n it represents an individual feature. In contrast to the previous case of pixel-based feature, which equally applies, spatial and/or temporal information can also be computed depending on the spatial and temporal interaction of the element to its neighborhood as in edge, texture and motion features. To completely exploit their potential, the spatial and/or tempo- ral properties of these features need to be taken into account in all the background subtraction steps to be fully addressed. Practically, these block-based features can be assigned to a central pixel of a block (or neighborhood), or to all the block. For example, textures such as Local Binary Pattern can be assigned at each central pixel of a block size 3 3 by moving this block all over the frame, or to all the block as in the works of Heikkila and Pietikainen [30], and Heikkila et al. [31] which used a pixel-wise LBP histogram based one (LBP-P) and a block-wise LBP histogram based approach (LBP-B), respectively. Thus, the block-based features can be used in pixel-wise or block-wise background modeling and foreground detection. When the features are obtained from the video compressed domain, the approach is mandatory block-based because the block are pre-defined and thus they cannot be moved over the frame. However, in block-based modeling and comparison, blocks (also called patches [32– 35]) can overlap or not [36]. A block is usually obtained as a vector of 3 3 neighbors of the current pixel. The advantage is to take into account the spatial dimension to improve the robustness and to reduce the computation time. Further- more, blocks can be of spatio-temporal type called spatio- temporal blocks [37], spatio-temporal neighborhoods [38], spatio-temporal patches [39–41] or bricks [42–44]) that in- trinsically encapsulate temporal information within spatial relationships of a group of pixels. In Pokrajac and Late- cki [37], a dimensionality reduction technique is applied to obtain a compact vector representation for each block. These blocks provide a joint representation of texture and motion patterns. One advantage is their robustness to noise and to the movement in the background. However, the disadvan- tage is that the detection is less precise because only blocks are detected, making them unsuitable for applications that require detailed shape information.

Region-based Features: Region-level (cluster-level, super-pixel-level) features consider element sizes that are non- uniform across the image frame considered, and then spe- cific features are computed on the corresponding element size. First, pixels in an image frame are grouped using an application-specific homogeneity criteria, typically exploit- ing partitioning mechanisms as follows: (1) region-based mechanisms as in Lin et al. [23] with the notion of Regions of Difference (RoD), (2) shape mechanisms as proposed in Jacobs and Pless [24], (3) behavior mechanisms as in Jodoin et al. [25], (4) clustering mechanisms as discussed by Bhaskar et al. [22,45,46], and Park and Byun [47], and (5) super-pixel mechanisms as in Sobral et al. [26], Ebadi et al. [48,49], Zhao et al. [50] and Chen et al. [51]. For example in Bhaskar et al. [22], each cluster contains pixels that have similar features in the color space. Then, the background model is applied on these clusters to obtain cluster of pixels classified as background or foreground. This cluster-wise approach gives less false alarms. Instead of the block-wise approach, the foreground detection is obtained at a pixel- level precision.

Pixel-based features need less time to be extracted than block-based or region-based features which require to be computed. In literature, in general, it can be summarized that the size of the feature and the comparison element determines the robustness of background modeling to noise and the challenges met in the videos, and often controls precision of foreground detection. A pixel-based modeling and comparison gives a pixel-based pre- cision but it is less robust to noise compared to block-based or region-based based modeling and comparison. However, there are several works which combined block-based (or region-based) and pixel-based approaches to reduce computation time by first using a block (or region) approach, and second to obtain a pixel precision by using a pixel-based approach, and they can be classified as follows: (1) multi-scales strategies [52–57], (2) multi-levels strate- gies [58–68], (3) multi-resolutions strategies [69–72], (4) multilayers strategies [41,73–84], (5) hierarchical strategies [85–94], and (6) coarse-to-fine strategies [95–99]. The analysis of these different approaches is out of the scope of this review, and the reader can found details about these strategies in [14].

III. CLASSIFICATION BY TYPE

Features can be computed in the pixel domain or in a transform domain. In this section, features those are predominantly com- puted in each domain and their robustness to critical situations in real videos, are discussed.

Features in the pixel domain

Features are popularly computed in the pixel domain as the value of the pixel is directly available. The following features are commonly used:

Intensity features: Intensity features are the most basic features that can be provide by gray-level cameras or infra- red (IR) cameras (see Section 3).

Color features: The color features in the RGB color space are most widely used because it is directly available from the sensor or the camera. But the RGB color space has an impor- tant drawback: its three components are dependent to each other which increases its sensitivity to illumination changes. For example, if a background point is covered by the shadow, the three component values at this point could be affected because the brightness and the chromaticity information are not separated. Thus, the three component values increase or decrease together as the lighting increases or decreases, re- spectively [100]. Alternative color spaces that have also been explored in the literature include YUV or YCrCb spaces. Sev- eral liminating the chances of leaving ghosts when foreground objects begin to move. Despite some compelling advantages, edge features (high pass filters) tend to vary more than other compara- ble features based on low pass filters [100]. For example, edge features in the horizontal and vertical directions have different reliability characteristics, since textured objects have high values in both directions, whereas homogeneous objects have low values in both directions (see Section 5).

Texture features: Texture features are appropriate to cope with illumination changes and shadows. Some common texture features that are generally used within this domain include the Local Binary in color [115– 121] (see Section 7).

Motion features: Motion features are usually obtained via optical flow but with the limitation of the computational time. Motion features allow the model to deal with irrel- evant background motion and clutter [122–128] (see Sec- tion 8).

Local histogram features: Local histograms are usually computed on color features [129–138]. But, local histograms can also be computed on edge features [83,139–142] to ob- tain Histograms of Oriented Gradients (HOG) (see Section 9).

Local histon features: Histon [143] is a contour plotted on the top of the histograms of three primary color components of a region in a manner that the collection of all points falling under the similar color sphere of predefined radius, called similarity threshold, belongs to one single value. The similar color sphere is the region in RGB color space such that all the colors falling in that region proposed to use other features like edge, texture and stereo features in addition to the color features.

computation time due to their size of 2563 2563 in RGB, and 256 2563 in gray level. Hence, the single channel is quantized to a finite number of levels l. Due to this, the correlograms' size is further reduced to l l with l 256. Correlogram can be extended to fuzzy correlogram [145] and multi-channel fuzzy correlogram [146] (see Section 11).

Haar-like features: Some authors [94,147–149], used the Haar-like features [19]. Haar-like features are features de- fined in real-time face detector and based on the similarity with Haar wavelets. Haar-like features are computed from adjacent rectangular areas at a given location in a detection window by adding the pixel intensities in each area and by calculating the difference between these sums. The main advantage of Haar-like features is their computation speed. With the use of integral images, Haar-like features of any size can be computed in constant time (see Section 12). in presence of gradual or sudden illumination changes [359]. Then, different strategies can be found in literature to alle- viate the limitations of comparisons between these color spaces are available in the literature including [101–105] and usually YCrCb is selected as the most appropriate color space. Although color features are often very discriminative features of objects, they have several limitations in the presence of challenges such as illumination changes, camouflage and shadows (see Edge features: The ambient light present in the scene can significantly affect the appearance of moving objects. How- ever, spectral features, are limited by their ability to adapt to such changes in appearance. Thus, edge features emerged as a robust alternative for moving object detection. Edge fea- tures are generally computed using a gradient approaches such as Canny, Sobel [100,106–111] or Prewitt [112,113] edge detector. It is commonly believed that edge features can handle local illumination changes, thus e

Pattern (LBP) [31], and the Local Ternary Pattern (LTP) [114]. Numerous variants of LBP and LTP can be found in the literature as can be seen summarized in Table 5. Furthermore, statistical and fuzzy textures can be used as developed in Section 6.

Stereo features: The extraction of stereo features rely on the need and use of specific acquisition systems such as a stereo, 3D, multiple, Time of Flight (ToF) cameras or RGB-D cameras (Microsoft Kinect,2 or Asus Xtion Pro Live3) to obtain the disparity information that usually represent the depth in the visual scene. It has become well-known that stereo features allow the model to deal with the camouflage

can be classified as one color. For every intensity value in the base histogram, the number of pixels falling under similar color sphere is calculated, and this value is added to the histogram value to get the histon value of that intensity. Histon can be extended to 3D histon and 3D Fuzzy histon as developed by Chiranjeevi and Sengupta [143] (see Section 10).

Local correlogram features: Correlogram was originally proposed for computer vision applications like object track- ing [144]. Since, correlogram captures the inter-pixel rela- tion of two pixels at a given distance, spatial information is obtained in addition to the color information. Thus, correlo- grams can efficiently alleviate the drawbacks of histograms, which only consider the pixel intensities for calculating the distribution. The main drawback of correlograms is their Section 4). In order to solve such issues, authors have also

the basic color spaces: (1) the use of well-known color spaces which separate the luminance and the chrominance information such as HSV and YCrCb,

(2) the use of designed shape color space models such as the cylinder color model [235,237,360,361], the hybrid cone- cylinder [236,362], the ellipsoidal color model [238], the box-based color model [239], and the double-trapezium cylinder model [242], (3) the use of characteristics in ad- dition of the intensity or color value (mean, variance, min- imum, maximum, etc.) (see Section 16), (4) the use of de- signed illumination invariant intensity or color features ob- tained by normalization [205,219,243,363], (5) the use of illumination compensation methods [364–373], and (6) the addition of other features (see Section 17). Normalization based features sacrifice discriminability while texture fea- tures cannot operate on texture-less regions. Both types of features produce large missing regions in the foreground mask.

Edge features: Edge features are obtained with edge de- tectors which operate on the difference between neigh- boring pixels, hence an edge detector should be reason- ably insensitive to global shifts in the mean level, i.e. to global illumination changes. Therefore it is interesting to run background/foreground separation algorithms on the output from edge detectors, hopefully reducing the effects of rapid illumination changes. So, the edge could handles the local illumination changes but also the ghost leaved when waking foreground objects begin to move. However, edge features are not sufficiently good to segment the foreground objects isolatedly. Indeed, edge features can sometimes handle dark and light camouflage problems and it is less sensitive to global illumination changes than color feature [111]. Nevertheless, problems like noise, false negative edges due to local illumination problems, foreground aperture and camouflage do not allow an accurate foreground detection. Furthermore, due to the fact that it is sometimes difficult to segment the foreground object borders, it is not possible to fill the objects, and solve the foreground aperture problem. Since it is not possible to handle dark and light camouflage problems only by using edges due to the foreground aperture difficulty, the brightness of color model is used to solve this problem and help to fill the foreground objects.

(gradient). As the gradient is less sensitive to illumination changes, the two types of feature vectors are integrated under the Bayes framework in the basic product formulation of the likelihoods.

– Features for dynamic background pixels: For modeling dynamic background pixels associated with non stationary objects, color co-occurrences are used as their dynamic features. This is because the color co-occurrence between consecutive frames has been found to be suitable to describe the dynamic features associated with non stationary background objects, such as moving tree branches or a flickering screen.

Features and strategies

There are several strategies in literature such as multi-scales strategies, multi-levels strategies, multi-resolutions strategies, multi-layers strategies, hierarchical strategies, and coarse-to-fine strategies (see Section 2.1). Practically, different features can be used following the scale, the level or the resolution. For example, a feature can be used at the block level (such as Haar-like features in [94]), and other features can be used at the pixel level (such as RGB in [94]). Thus, these strategies employed multiple features schemes. Please see Tables 9–11 for a quick overview.

Features and similarities

The foreground mask is usually obtained from a similarity/dissimilarity measure between (1) the direct value of the feature in the background model and the current frame, or (2) a value computed from the direct value of the feature (mean, variance, probability, etc...) in the background model and the current frame. This value can be a scalar (intensity value, mean, probability, etc.), a vector (2D spatial vector, 3D spatiotemporal vector, etc...) or a histogram (correlogram, etc.). Practically, comparison of features can be made by using similarity/dissimilarity measures obtained with

(1) a crisp, statistical or fuzzy distance for scalar cases, (2) a ratio for

scalar cases, (3) linear dependence measure for vector cases, and

a intersection measure for histogram (correlogram) case. The choice of the suitable similarity/dissimilarity measure is guided by the properties and the distribution of the concerned features. Furthermore, spatial and temporal features such as LBP and LTP need also measures for their computing as follows: (1) a measure for the distance in the spatial neighborhood, and (2) a measure for

Texture features: Texture features allow to be robust in presence of shadows and gradual illumination changes, and sometimes in dynamic backgrounds. Texture features can produce false detections due to textures induced by local illumination effects like in cast shadows. Furthermore, an algorithm based only on texture may cause detection errors in regions of blank texture and heterogeneous texture.

Motion features: Motion features can handle irrelevant background motion and clutter such as waving trees and waves.

Stereo features: Stereo features allow the model to deal with the camouflage in color but not in depth.

Thus, multiple features approaches with two, three or a set of features obtained from a bag-of features or by feature selection are suitable to address multiple challenges in the same video (see Section 17). A representative work developed by Li et al. [109] consists in a sets of features built following the type of background (static or dynamic) as follows:

Features for static background pixels: For modeling pixels belonging to a stationary background object, the stable and most significant features are its color and local structure

the distance in the temporal neighborhood. Thus, for spatial and temporal features like texture, it needs to choose three distances. We list below the different similarity/dissimilarity measures used in the literature for foreground detection (see Table 8 for a quick overview):

Similarities for scalar case: Scalar value is the most common case in the literature and the similarities used can be classified as follows:

– Difference: The difference computed in a pixel-wise manner between the feature in the background model and the current frame is the most measure used. So, the difference is obtained by a distance and then a threshold is used to classify the pixel as background or foreground as follows:

$$\text{distance}(B(x, y) - I(x, y)) < T \qquad (4)$$

where B(x, y) and I(x, y) are the values of the feature in the background image and in the current image, respectively. distance(,) is a distance function. Several distance functions have been used in the literature and they can be classified as follows:

Crisp distance: The most common distance function used for intensity/color values is the absolute distance [221,374]. Aach et al. [375] used a total least squares distance measure. In an other work, Yadav and Sing used a quasi-euclidean distance. To compare Spatiotemporal Condition Information (SCI), Wang et al. [38] designed a specific measure called Neighborhood Weighted Spatiotemporal Condition Information (NWSCI). Using compressive features [376], Yang et al. [377] developed a (Pixel-to-Model) P2M distance.

Statistical distance: To compare the K distribution in the original MOG, Stauffer and Grimson [20] used the Mahalanobis distance with the RGB features. An alternative to the Mahalanobis distance is the Kullback–Leibler (KL) divergence used in Makantasis et al. [378] with the infrared features and Patwardhan et al. [379] with the RGB features. In a further work, Pavlidis et al. [380] claimed that the MOG algorithm needs a divergence measure between two distributions so that if the divergence measure between the new distribution and one of the existing distributions is ‘‘too small’’, these two distributions could be merged together. Thus, Pavlidis et al. [380] used the Jeffreys divergence measure to check if the incoming pixel value can be ascribed to any of the existing K Gaussians. Experimental results presented by Pavlidis et al. [380] show that the false positives are reduced in comparison with the Mahalanobis distance and the KL divergence. In an other work, Santoyo-Morales and Hasimoto-Beltran used the Chi-2 distance with YUV features instead of the Mahalanobis distance. In a non parametric model based on KDE, Ko et al. [381] choose the Bhattacharyya distance due to its low computational cost. In an other work, Mukherjee et al. [83] developed a distance measure based on support weight to compare RGB features. St-Charles and Bilodeau [382] employed the Hamming distance to compare LSBPs.

Order-Consistency Measure: Xie et al. [189] used an explicit model for the camera response function, the camera noise model, and illumination prior. Assum- ing a monotone and nonlinear camera response func- tion, Xie et al. [189] show that the sign of the difference between two pixel measurements is maintained across global illumination changes. Noise statistics are used to transform each frame into a confidence frame where each pixel is replaced by a probability that it is likely to keep its sign with respect to the most different pixel in its neighborhood. Hence, an order consistency measure is defined as a distance between two distri- butions. Xie et al. [189] used the Bhattacharyya dis- tance due to its properties to the Bayes error. Finally, an Illumination Invariant Change Detector via order consistency (IICD-OC) is developed. Experimental results [189] on videos taken by an omni-directional camera show the robustness of IICD-OC against illu- mination changes. But, the ordinal measure required a reordering of blocks and it is computationally ex- pensive. To solve this problem, Singh et al. [383] ex- plicitly modeled noise under which rank-consistency is tested, and used a probabilistic generative model under which frame blocks are generated. The order- consistency is posed as a hypothesis validation prob- lem using fast significance testing based on PAV. In a further work, Parameswaran et al. [373] used the same order-consistency measure in an illumination compensation approach

Location features: The location (x, y) can be used as a fea- ture to exploit the dependency between the pixel [150–154] (see Section 14).

1.1.1. Feature relevance and learning

To choose the most discriminative features in a multiple features or feature selection scheme, feature relevance may be addressed. More generally, feature relevance can be determined in feature learning scheme which can be classified as developed in Zhong et al. [348]:

1. Traditional feature learning: This category includes linear algorithms and their kernel extension, and manifold learning method. Practically, an learning algorithm can be linear or nonlinear, supervised or unsupervised, generative or discriminative, global or local. For example, Principal Compo- nent Analysis (PCA) is a linear, unsupervised, generative and global feature learning method, while Linear Discriminant Analysis (LDA) is a linear, supervised, discriminative and global method. Global methods aim to preserve the global information of data in the learned feature space, but local ones focus on preserving local similarity between data dur- ing learning the new representations. For instance, unlike PCA and LDA, Locally Linear Embedding (LLE) is a locality- based feature learning algorithm. Locality-based feature learning like LLE as manifold learning, since it is to discover the manifold structure hidden in the high dimensional data.
2. Deep learning algorithms: Deep learning models includes models like Convolutional Neural Network (CNN) [349] and Recurrent neural network (RNN). A survey of deep learning models can be found in Schmidhuber [349].

Feature relevance has been less investigated in background modeling and foreground detection methods than manual im- age feature methods, such as Local Binary Patterns (LBP) [31], histogram of oriented gradients (HOG) [139], and Scale-Invariant Feature Transform (SIFT) [252]. For traditional feature learning, the one work which concerns feature relevance is the work of Molina-Giraldo et al. [350,351]. The feature relevance analysis is made through a Principal Component Analysis (PCA), searching for directions with greater variance to project the data. Thus, the relevance of the original features is quantified with weight- ing factors. Finally, Molina-Giraldo et al. [350,351] developed a background subtraction method based a multi-kernel learning in which the weight are selected from the feature relevance analysis. Experimental results [350,351] on the I2R dataset [109] show that the proposed Weighted Gaussian Kernel Video Segmentation (WGKVS) model outperforms SOBS [352]. For deep learning algo- rithms, the approaches available in literature can be classified as follows: (1) Deep Auto-encoder Networks (DAN) [16,353,354], (2) Convolutional Neural Networks (CNN) [17,18,355,356], (3) Neural Response Mixture (NeREM) [357].

1.1.2. Features and challenges

In this section, we grouped all the advantages and disadvantages of the different features in terms of robustness against the different challenges met in video and detailed in Bouwmans [9], and they can be summarized as follows:

– Color features: Although intensity and color features are often very discriminative features and allow basic foreground detection, they are not robust in challenges such as illumination changes, foreground aperture, camouflage in color and shadows. However, intensity can be used in complementarity of color to deal with different color problems such as dark foreground and light foreground. Furthermore, this combination solves saturation problems and minimum intensity problems [358], and reduces the number of false negatives, false positives and increase true positives. But, the intensity as colors cannot work with intense shadows and highlight that often occur in indoor and outdoor scenes.

3. Conclusion

In conclusion, this review on the role and the importance of features for background modeling and foreground detection highlights the following points:

– Features can be classified following their size, their type in a specific domain, their intrinsic properties and their mathematical concepts. Each type of features presents different robustness against challenges met in videos taken by a fixed cameras. For the color feature, YCrCb color space seems to be the more suitable feature [105,384]. For the texture feature, Silva et al. [339] provided a study on the LBP and its variants that show that XCS-LBP is the best LBP feature for this application in presence of illumination changes and dynamic backgrounds. Although this study covered texture features, it is restricted to LBP features and then there is not a full study on the different texture features. For the depth feature, it needs to carefully used them following their properties as developed in Nghiem and Bremond [560]. Features in a domain transform are very useful to reduce computation times as in the case of compressive sensing features.

– Several features have been used in other applications and none in background modeling and foreground detection such several variants of LBP (Multi-scale Region Perpendicular LBP (MRP-LBP) [299], Scale- and Orientation Adap- tive LBP (SOA-LBP) [300]). Furthermore, statistical or fuzzy version of crisp feature could be investigated such as his- tograms of fuzzy oriented gradients [202]. It would be in- teresting to evaluate them for this application.

– Because each feature has its strengths and weaknesses against each challenge, multiple features schemes are used to combine the advantages of their different robustness. Most of the time, gradient, texture, motion and stereo fea- tures are used in addition to the color feature to deal with camouflage in color, illumination changes, dynamic back- grounds and shadows. Different fusion operators can be used to combine these different features but fuzzy integrals such as the Choquet integral [330] and interval-valued Cho- quet [188] seem the best way to aggregate different features because dependency between features can be taken into account. Because there is not a unique feature that performs better than any other feature independently of the background and foreground properties, feature selection allows to use the best feature or the best combination of features. Exper- imental results provided by the existing approaches show the pertinence of feature selection in background modeling and foreground detection. However, basic algorithms such as Adaboost and Realboost have been used most of the time. The most advanced scheme is the IWOC-SVM algorithm developed by Silva et al. [339], but more advanced selection schemes can be used such as statistical or fuzzy feature selection.

To summarize, the most interesting approach seems to fuse mul- tiple features with the intervalued fuzzy Choquet integral. The best set of features seems to be illumination invariant color fea- tures combined with spatio-temporal texture features and depth features. Future research should concern (1) a full evaluation of texture features, (2) a full comparison of feature fusion schemes,
(1) feature selection schemes and (4) reliability of features because it has been less investigated. Finally, features learned by deep learning methods such as Stacked Denoising Auto-Encoder (SDAE) [16] and Convolutional Neural Networks (CNN) [17,18] are surely the features that will outperforms all the other features because deep learning methods have the sole ability of learning features that best fit a given set of data. Furthermore, unlike conventional hand-crafted features, learned features come from multiple layers which focus on various level of details in the video. Thus, learned feature representation allows to well capture the intrinsic structural properties of a scene and adaptively discover a set of filter patterns that are robust to complicated factors such as noise and illumination variation.

1. References

[1] S. Cheung, C. Kamath, Robust background subtraction with foreground vali- dation for urban traffic video, EURASIP J. Appl. Signal Process. (2005).

[2] J. Carranza, C. Theobalt, M. Magnor, H. Seidel, Free-viewpoint video of human actors, ACM Trans. Graph. 22 (3) (2003) 569–577.

[3] F. El Baf, T. Bouwmans, B. Vachon, Comparison of background subtraction methods for a multimedia learning space, in :International Conference on Signal Processing and Multimedia, SIGMAP 2007, July 2007.

[4] A. Vacavant, T. Chateau, A. Wilhelm, L. Lequievre, A benchmark dataset for foreground/background extraction, in: International Workshop on Back- ground Models Challenge, ACCV 2012, November 2012.

[5] T. Bouwmans, A. Sobral, S. Javed, S. Jung, E. Zahzah, Decomposition into low- rank plus additive matrices for background/foreground separation: A review for a comparative evaluation with a large-scale dataset, Comput. Sci. Rev. (2016).

[6] T. Bouwmans, F.E. Baf, B. Vachon, Background modeling using mixture of gaussians for foreground detection - A survey, Recent Pat. Comput. Sci. RPCS 2008 1 (3) (2008) 219–237.

[7] T. Bouwmans, Subspace Learning for Background Modeling: A Survey, Recent Patents on Computer Science, RPCS 2009 2 (3) (2009) 223–234.

[8] T. Bouwmans, Recent advanced statistical background modeling for fore- ground detection: A systematic survey, Recent Pat. Comput. Sci. RPCS 2011 4 (3) (2011) 147–176.

[9] T. Bouwmans, Traditional and recent approaches in background modeling for foreground detection: An overview, Comput. Sci. Rev. 11 (2014) 31–66.

[10] T. Bouwmans, E. Zahzah, Robust PCA via principal component pursuit: A review for a comparative evaluation in video surveillance, Comput. Vis. Image Underst. 122 (2014) 22–34 Special Isssue on Background Models Challenge.

[11] T. Bouwmans, F. El Baf, B. Vachon, Statistical Background Modeling for Foreground Detection: A Survey, in: Part 2, Chapter 3, Handbook of Pattern Recognition and Computer Vision, vol. 4, World Scientific Publishing, Prof C.H. Chen, 2010, pp. 181–199.

[12] T. Bouwmans, Background subtraction for visual surveillance: A fuzzy ap- proach, in: S.K. Pal, A. Petrosino, L. Maddalena (Eds.), Handbook on Soft Computing for Video Surveillance, Taylor and Francis Group, 2012, pp. 103– 139 (Chapter 5).

[13] C. Guyon, T. Bouwmans, E. Zahzah, Robust principal component analysis for background subtraction: Systematic evaluation and comparative analysis, in: Principal Component Analysis, INTECH, 2012, pp. 223–238 (Book 1, Chap- ter 12).

[14] T. Bouwmans, F. Porikli, B. Hoferlin, A. Vacavant, Handbook on Background Modeling and Foreground Detection for Video Surveillance, Chapman and

Hall/CRC, 2014.

[15] T. Bouwmans, J. Gonzalez, C. Shan, M. Piccardi, L. Davis, Special issue on back- ground modeling for foreground detection in real-world dynamic scenes, Mach. Vis. Appl. (2014) (special issue).

[16] Y. Zhang, X. Li, Z. Zhang, F. Wu, L. Zhao, Deep learning driven blockwise mov- ing object detection with binary scene modeling, Neurocomputing (2015).

[17] M. Braham, M. Van Droogenbroeck, Deep background subtraction with scene-specific convolutional neural networks, in: International Conference on Systems, Signals and Image Processing, IWSSIP 2016, 2016.

[18] Y. Wang, Z. Luo, P. Jodoin, Interactive deep learning method for segmenting moving objects, Pattern Recognit. Lett. (2016) Special Issue on Scene Back- ground Modeling and Initialization.

[19] P. Viola, M. Jones, Rapid object detection using a boosted cascade of simple features, Comput. Vis. Pattern Recognit. (2001).

[20] C. Stauffer, E. Grimson, Adaptive background mixture models for real-time tracking, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 1999, 1999, pp. 246–252.

[21] X. Fang, W. Xiong, B. Hu, L. Wang, A moving object detection algorithm based on color information, in: International Symposium on Instrumentation Science and Technology, vol. 48, 2006, pp. 384–387.

[22] H. Bhaskar, L. Mihaylova, A. Achim, Video foreground detection based on symmetric alpha-stable mixture models, IEEE Trans. Circuits Syst. Video Technol. (2010).

[23] Y. Lin, Y. Tong, Y. Cao, Y. Zhou, S. Wang, Visual-attention based background modeling for detecting infrequently moving objects, IEEE Trans. Circuits Syst. Video Technol. (2016).

[24] N. Jacobs, R. Pless, Shape background modeling : The shape of things that came, in: IEEE Workshop on Motion and Video Computing, WMVC 2007, 2007, pp. 1–7.

[25] P. Jodoin, V. Saligrama, J. Konrad, Behavior subtraction, IEEE Trans. Image Process. (2011).