5

Background Subtraction for Visual Surveillance: A Fuzzy Approach

Thierry Bouwmans
University of La Rochelle, La Rochelle, France

5.1 Introduction

Analysis and understanding of video sequences is an active research field. Many applications in this research area (video surveillance [18], optical motion capture [2], multimedia application [16]) need in the first step to detect the moving objects in the scene. So, the basic operation needed is the separation of the moving objects, called the foreground, from the static information, called the background. The process mainly used is background subtraction [12, 14, 24]. The simplest way to model the background is to acquire a background image that does not include any moving object. In some environments, the background is not available and can always be changed un-

der critical situations like illumination changes, or objects being introduced or removed from the scene. So, the background representation model must be more robust and adaptive. The different background representation models can be classified into the following categories:

- ***Basic background modeling:*** In this case, the background is modeled using an average [11], a median [42], or a histogram analysis over time [64]. Once the model is computed, pixels of the current image are classified as foreground by thresholding the difference between the background image and the current frame as follows:

$$d\left(I_t(x,y), B_{t-1}(x,y)\right) > T. \tag{5.1}$$

  Otherwise, pixels are classified as background. T is a constant threshold; $I_t(x,y)$, $B_{t-1}(x,y)$ are respectively the current image at time t; and the background image at time $t-1$. $d(.,.)$ is a distance measure that is usually the absolute difference between the current and the background images.

- ***Statistical background modeling:*** The background is represented using a single Gaussian [61], a Mixture of Gaussians [57], or a Kernel Density Estimation [23]. Statistical variables are used to classify the pixels as foreground or background.

- ***Background modeling based on clusters:*** The background model supposes that each pixel in the frame can be represented temporally by clusters. Incoming pixels are matched against the corresponding cluster group and are classified according to whether the matching cluster is considered part of the background. The clustering approach consists of using the K-mean algorithm [15] or using Codebook [34].

- ***Neural network background modeling:*** The background is represented by the mean of the weights of a neural network suitably trained on N clean frames. The network learns how to classify each pixel as background or foreground [19, 38].

- ***Background estimation:*** The background is estimated using a filter. Any pixel of the current image that deviates significantly from its predicted value is declared foreground. This filter may be a Wiener filter [60], a Kalman filter [43], or a Tchebychev filter [17].

All these different models present the same following steps and issues:

- Background modeling, which describes the kind of model used to represent the background
- Background initialization, which concerns the initialization of the model
- Background maintenance, which relies on the mechanism used for adapting the model to the changes occurring in the scene over time

- Foreground detection which consists of the classification of the pixel as a background or as a foreground pixel
- Choice of the picture's element, which is used in the previous steps; this element may be a pixel [57], a block [25, 46], or a cluster [9, 10]
- Choice of the features which characterize the picture's element. In the literature, there are five features commonly used: color features, edge features, stereo features, motion features, and texture features. In [37], these features are classified as spectral features (color features), spatial features (edge features, texture features), and temporal features (motion features). These features have different properties that allow one to handle differently the critical situations (illumination changes, motion changes, structure background changes)

Figure 5.1 shows an overview of a background subtraction process.

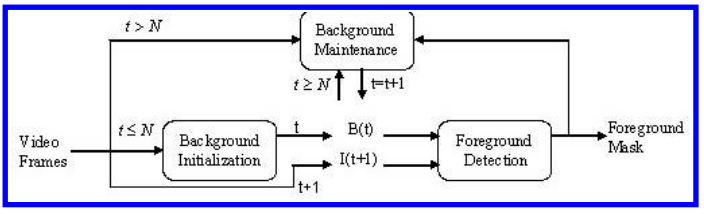


FIGURE 5.1 Background subtraction process.

In developing a background subtraction method, researchers must design each step and choose the features in relation to the critical situations [60] they want to handle: noise image due to a poor quality image source, camera jitter, camera automatic adjustments, time of the day, light switch, bootstrapping, camouflage, foreground aperture, moved background objects, inserted background objects, multimodal background, waking foreground object, sleeping foreground object, and shadows. These critical situations have different spatial and temporal properties. The main difficulties come from the illumination changes and dynamic backgrounds:

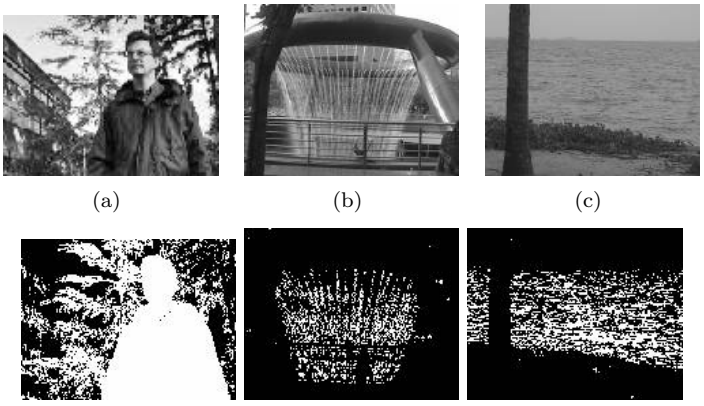
- Illumination changes appear in indoor and outdoors scenes. Figure 5.2 shows an indoor scene that presents a gradual illumination change. It causes false detections in several parts of the foreground mask as can be seen in Figure 5.2-d). Figure 5.3 shows the case of a sudden illumination change due to a light on/off. As all the pixels are affected by this change, a large number of false detections are generated (see Figure 5.3-c)).
- Dynamic backgrounds appear in outdoor scenes. Figure 5.4 shows three main types of dynamic backgrounds: waving trees, water rippling, and water surface. The first row shows the original images and the second row shows the foreground mask obtained by the GMM [57]. In each case, there are a large number of false detections.

(a) (b) (c) (d)

FIGURE 5.2 a) An indoor scene with low illumination; b) the same scene with a moderate illumination; c) the same scene with a high illumination; d) the foreground mask obtained with GMM [57]. This sequence, called "Time of Day", comes from the Wallflower dataset [60].

(a) (b) (c)

FIGURE 5.3 a) An indoor scene with light-on; b) the same scene with light-off; c) the foreground mask obtained with GMM [57]. This sequence, called "Light Switch", comes from the Wallflower dataset [60].



(a) (b) (c)

FIGURE 5.4 The first row presents original scenes containing dynamic backgrounds: a) waving trees [60], b) fountain [37] and c) water surface [37]. The second row shows the corresponding foreground masks obtained by the GMM [57].

All these critical situations generate imprecision and uncertainties in the whole process of background subtraction. Therefore, some authors have recently introduced fuzzy concepts in the different steps of background subtraction as follows:

- ***Fuzzy Background Modeling:*** The main challenge consists of modeling a multimodal background. The algorithm usually used is the Gaussian Mixture Models [61]. The parameters are determined using a training sequence which contains insufficient or noisy data. So, the parameters are not well determined. In this context, Type-2 Fuzzy Gaussian Mixture Models [6, 7, 13] are used to model uncertainties when dynamic backgrounds occur.

- ***Fuzzy Foreground Detection:*** In this case, a saturating linear function is used to avoid crisp decision in the classification of the pixels as background or foreground. The background model can be unimodal, such as the running average in [54, 55], or multi-modal, such as the background modeling with confidence measure proposed in [50]. Another approach consists of aggregating different features such as color and texture features using the Sugeno integral [63] or the Choquet integral [3–5]. Fuzzy foreground detection is more robust to illumination changes and shadows than crisp foreground detection.

- ***Fuzzy Background Maintenance:*** The idea is to update the background following the membership of the pixel at the class background or foreground. This membership comes from the fuzzy foreground detection. This fuzzy adaptive background maintenance allows one to deal robustly with illumination changes and shadows.

- ***Fuzzy Post-Processing:*** Fuzzy inference can be used between the previous and the current foreground masks to perform detection of the moving objects, as developed recently by Sivabalakrishnan and Manjula [56].

TABLE 5.1 Fuzzy background subtraction: An overview.

Background Subtraction Steps	Algorithm	Authors - Dates
Background Modeling	Type-2 Fuzzy GMM (3)	El Baf et al. (ISVC 2008) [6]
Foreground Detection	Saturating Linear Function(2)	Sigari et al. (IJCSNS 2008) [55]
	Saturating Linear Function(2)	Shakeri et al. (ICSP 2008) [52]
	Saturating Linear Function(1)	Rossel et al. (ICPR 2010) [50]
	Sugeno Integral (1)	Zhang and Xu (FSKD 2006) [63]
	Choquet Integral (3)	El Baf et al. (WIAMIS 2008) [3]
	Choquet Integral (1)	Ding et al. (ComSIS 2010) [22]
	Choquet Integral (1)	Azab et al. (ICIP 2010) [1]
Background Maintenance	Fuzzy Learning Rate (2)	Sigari et al.(IJCSNS 2008) [55]
	Fuzzy Learning Rate (3)	Maddalena et al.(WILF 2009) [39]
	Fuzzy Maintenance Rule (1)	El Baf et al.(ICIP 2008) [8]
	Fuzzy Maintenance Rule (1)	Kashani et al. (IJCTE 2010) [33]

Table 5.1 shows an overview of the fuzzy concepts used in the background subtraction. The first column indicates the background subtraction steps. The second column indicates the fuzzy concepts used with the papers counted for each method in the parenthesis. The third column gives the name of the authors and the date of their first publication that use the corresponding fuzzy concept.

The rest of this chapter is organized as follows. In the Section 5.2, we present the fuzzy background modeling method using Type-2 Fuzzy Gaussian Mixture Models. Then, in Section 5.3, fuzzy foreground detection methods are described. In Section 5.4, we present how some authors used the results of the fuzzy foreground detection to update the background. Finally, we present several experimental results that show the performance of the fuzzy methods versus the traditional methods.

5.2 Background Modeling by Type-2 Fuzzy Gaussian Mixture Models

Gaussian Mixture Models (GMMs) [61] are the most popular techniques to deal with dynamic backgrounds but present some limitations when some rapid dynamic changes occur, like camera jitter, illuminations changes, and movement in the background. Furthermore, the GMMs are initialized by an expectation-maximization (EM) algorithm that allows one to estimate GMMs parameters from a training sequence according to the maximum-likelihood (ML) criterion. The GMMs are completely certain once their parameters are specified. However, because of insufficient or noisy data in the training sequence, the GMMs may not accurately reflect the underlying distribution of the observations according to the ML estimation. It may seem problematical to use likelihoods that are themselves precise real numbers to evaluate GMMs with uncertain parameters. To solve this problem, Type-2 Fuzzy Gaussian Mixture Models (T2-FGMMs) were developed by Zeng et al. [62] to introduce descriptions of uncertain parameters in the GMMs. T2-FGMMs have proved their superiority in pattern classification [62]. Recently, El Baf et al. [6, 7, 13] have proposed to model the background by using T2-FGMMs. Therefore, each pixel can be characterized by its intensity in the RGB color space. So, the observation o is a vector X_t in the RGB space with three components, that is $d = 3$. Then, the GMMs are composed of K mixture components of multivariate Gaussian as follows:

$$P(X_t) = \sum_{i=1}^{K} \omega_{i,t} \eta \left(X_t, \mu_{i,t}, \sum_{i,t} \right), \tag{5.2}$$

where K is the number of distributions, $\omega_{i,t}$ is a weight associated with the i^{th} Gaussian at time t with mean $\mu_{i,t}$ and standard deviation $\sum_{i,t}$. η is a Gaussian probability density function:

$$\eta(X_t, \mu, \sum) = \frac{1}{(2\pi)^{\frac{3}{2}} |\sum|^{\frac{1}{2}}} exp\left(-\frac{1}{2} (X_t - \mu)^T \sum^{-1} (X_t - \mu) \right). \tag{5.3}$$

For the T2-FGMM-UM, the multivariate Gaussian with uncertain mean vector is:

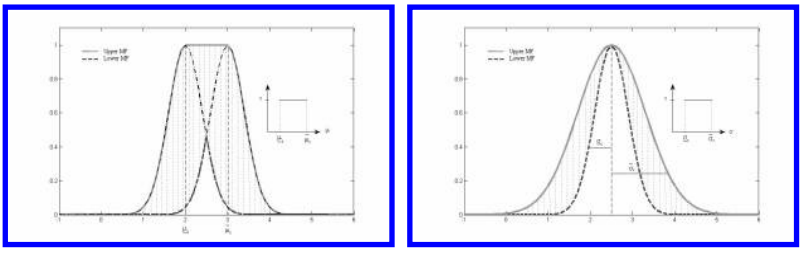
$$\eta\left(X_t, \tilde{\mu}, \sum\right) = \frac{1}{(2\pi)^{\frac{3}{2}}|\sum|^{\frac{1}{2}}} \prod exp\left[-\frac{1}{2}\left(\frac{X_{t,c}-\mu_c}{\sigma_c}\right)^2\right] \quad (5.4)$$

with $\mu_c \in \left[\underline{\mu}_c, \overline{\mu}_c\right]$ and $c \in \{R, G, B\}$.
For the T2-FGMM-UV, the multivariate Gaussian with uncertain variance vector is:

$$\eta\left(X_t, \mu, \tilde{\sum}\right) = \frac{1}{(2\pi)^{\frac{3}{2}}|\sum|^{\frac{1}{2}}} \prod exp\left[-\frac{1}{2}\left(\frac{X_{t,c}-\mu_c}{\sigma_c}\right)^2\right], \quad (5.5)$$

where $\sigma_c \in [\underline{\sigma}_c, \overline{\sigma}_c]$ and $c \in \{R, G, B\}$.

$\tilde{\mu}$ and $\tilde{\sigma}$ denote the uncertain mean vector and covariance matrix, respectively. Because there is no prior knowledge about the parameter uncertainty, practically Zeng et al. [62] assume that the mean and standard deviation vary within intervals with uniform possibilities, that is, $\mu \in \left[\underline{\mu}, \overline{\mu}\right]$ or $\sigma \in [\underline{\sigma}, \overline{\sigma}]$. Each exponential component in Equation (5.3) and Equation (5.4) is the Gaussian primary membership function (MF) with uncertain mean or standard deviation, as shown in Figure 5.5. The shaded region is the footprint of un-



(a) (b)

FIGURE 5.5 a) The Gaussian primary MF with uncertain mean; b) the Gaussian primary MF with uncertain std having uniform possibilities. The shaded region is the FOU. The thick solid and dashed lines denote the lower and upper MFs.

certainty (FOU). The thick solid and dashed lines denote the lower and upper MFs. In the Gaussian primary MF with uncertain mean, the upper MF is:

$$\overline{h}(o) = \begin{cases} f\left(o; \underline{\mu}, \sigma\right), & \text{if } o < \underline{\mu} \\ 1, & \text{if } \underline{\mu} \le o < \overline{\mu} \\ f\left(o; \overline{\mu}, \sigma\right), & \text{if } o > \overline{\mu}, \end{cases} \quad (5.6)$$

where $f\left(o; \underline{\mu}, \sigma\right) = exp\left[-\frac{1}{2}\left(\frac{o-\underline{\mu}}{\sigma}\right)^2\right]$ and $f\left(o; \overline{\mu}, \sigma\right) = exp\left[-\frac{1}{2}\left(\frac{o-\overline{\mu}}{\sigma}\right)^2\right]$.

The lower MF is:

$$\underline{h}(o) = \begin{cases} f(o; \overline{\mu}, \sigma), & \text{if } o \leq \frac{\underline{\mu}+\overline{\mu}}{2} \\ f\left(o; \underline{\mu}, \sigma\right), & \text{if } o > \frac{\underline{\mu}+\overline{\mu}}{2}, \end{cases} \tag{5.7}$$

In the Gaussian primary MF with uncertain standard deviation, the upper MF is $\underline{h}(o) = f(o; \mu, \overline{\sigma})$ and the lower MF is $\overline{h}(o) = f(o; \mu, \underline{\sigma})$.

The factors k_m and k_ν control the intervals in which the parameter varies as follows:

$$\underline{\mu} = \mu - k_m\sigma, \quad \overline{\mu} = \mu + k_m\sigma, \quad k_m \in [0, 3], \tag{5.8}$$

$$\underline{\sigma} = k_\nu\sigma, \quad \overline{\sigma} = \frac{1}{k_\nu}\sigma, \quad k_\nu \in [0.3, 1]. \tag{5.9}$$

Because a one-dimensional gaussian has 99.7% of its probability mass in the range of $[\mu - 3\sigma, \mu + 3\sigma]$, Zeng et al. [62] constrain $k_m \in [0, 3]$ and $k_\nu \in [0.3, 1]$. These factors also control the area of the FOU. The bigger k_m or the smaller k_ν, the larger the FOU, which implies the greater uncertainty.

Both T2-FGMM-UM and T2-FGMM-UV can be used to model the background, and we can expect that T2-FGMM-UM will be more robust than T2-FGMM-UV. Indeed, only the means are estimated and tracked correctly over time in the GMM maintenance. The variance and the weights are unstable and unreliable, as explained by Greiffenhagen et al. [28].

Training a T2-FGMM consists of estimating the parameters μ, $\sum$ and the factor k_m or k_ν. Zeng et al. [62] set the factors k_m and k_ν as constants according to prior knowledge. In background modeling, they are fixed depending on the video. Thus, parameter estimation of T2-FGMM includes three steps:

- Step 1: Choose K between 3 and 5.
- Step 2: Estimate GMM parameters by an EM algorithm.
- Step 3: Add the factor k_m or k_ν to GMM to produce T2-FGMM-UM or T2-FGMM-UV.

Once the training is made, a first foreground detection can be processed. By using the ratio $= r_j = \omega_j/\sigma_j$, we first order the K Gaussians as in [61]. This ordering supposes that a background pixel corresponds to a high weight with a weak variance due to the fact that the background is more present than moving objects and that its value is practically constant. The first B Gaussian distributions that exceed certain threshold T are retained for a background distribution:

$$B = argmin_b \left(\sum_{i=1}^{b} \omega_{i,t} > T \right). \tag{5.10}$$

The other distributions are considered to represent a foreground distribution. When the new frame arrives at times $t + 1$, a match test is made for each

pixel. For this, we use the log-likelihood, and thus we are only concerned with the length between two bounds of the log-likelihood interval, that is, $H(X_t) = \left| ln\left(\underline{h}(X_t)\right) - ln\left(\overline{h}(X_t)\right) \right|$. In Figure 5.5-a), the Gaussian primary MF with uncertain mean has:

$$H(X_t) = \begin{cases} \frac{2k_m|X_t-\mu|}{\sigma}, & \text{if } X_t \leq \mu - k_m\sigma \text{ or } X_t \geq \mu + k_m\sigma \\ \frac{|X_t-\mu|}{2\sigma^2} + \frac{k_m|X_t-\mu|}{\sigma} + \frac{k_m^2}{2}, & \text{if } \mu - k_m\sigma < X_t < \mu + k_m\sigma. \end{cases} \tag{5.11}$$

In Figure 5.5-b), the Gaussian primary MF with uncertain standard deviation has

$$H(X_t) = \left(\frac{1}{1/k_\nu^2 - k_\nu^2}\right)\frac{|X_t-\mu|^2}{2\sigma^2}. \tag{5.12}$$

μ and σ are the mean and the std of the original certain T1 MF without uncertainty. Both Equations (5.11) and (5.12) are increasing functions in terms of the deviation $|X_t - \mu|$. For example, given a fixed k_m, the farther the X_t deviates from μ, the larger $H(X_t)$ is in Equation (5.11), which reflects a higher extent of the likelihood uncertainty. This relationship agrees with the outlier analysis. If the outlier X_t deviates farther from the center of the class-conditional distribution, it has a larger $H(X_t)$, showing its greater uncertainty to the class model. So, a pixel is ascribed to a Gaussian if

$$H(X_t) < k\sigma, \tag{5.13}$$

where k is a constant threshold determined experimentally and is equal to 2.5. Then, two cases can occur: (1) A match is found with one of the K Gaussians. In this case, if the Gaussian distribution is identified as a background one, the pixel is classified as background, or else the pixel is classified as foreground. (2) No match is found with any of the K Gaussians. In this case, the pixel is classified as foreground. At this step, a binary mask is obtained. Then, to make the next foreground detection, the parameters must be updated.

T2-FGMM Maintenance is made as in the original GMM [61] as follows:

- Case 1: A match is found with one of the K Gaussians. For the matched component, the update is done as follows:

$$\omega_{i,t+1} = (1-\alpha)\,\omega_{i,t} + \alpha, \tag{5.14}$$

where α is a constant learning rate.

$$\mu_{i,t+1} = (1-\rho)\,\mu_{i,t} + \rho X_{t+1} \tag{5.15}$$

$$\sigma_{i,t+1}^2 = (1-\rho)\,\sigma_{i,t}^2 + \rho\,(X_{t+1}-\mu_{i,t+1})\,(X_{t+1}-\mu_{i,t+1})^T, \tag{5.16}$$

where $\rho = \alpha\eta(X_{t+1}, \mu_i, \sum_i)$.
For the unmatched components, μ and σ are unchanged, only the weight is replaced by $\omega_{j,t+1} = (1-\alpha)\,\omega_{j,t}$.

- Case 2: No match is found with any of the K Gaussians. In this case, the least probable distribution k is replaced by a new one with parameters:

$$\omega_{k,t+1} = \text{Low Prior Weight} \quad (5.17)$$
$$\mu_{k,t+1} = X_{t+1} \quad (5.18)$$
$$\sigma^2_{k,t+1} = \text{Large Initial Variance} \quad (5.19)$$

  Once a background maintenance is made, another foreground detection can be processed, and so on.

Results presented in Section 5.5.1 show the relevance of T2-FGMM in the presence of camera jitter, waving trees, and water rippling.

5.3 Foreground Detection

Foreground detection consists of classifying pixels as background and foreground by comparing the background and the current images. In general, a simple subtraction is made between these two images to detect regions corresponding to foreground. Another way to establish this comparison consists of defining a similarity measure between pixels in current and background images. In this case, pixels corresponding to background should be similar in the two images, while pixels corresponding to foreground should not be similar.

Another issue in the foreground detection is the choice of features. Color features are the main features used because colors are often very discriminative features of objects, but they do have several limitations in the presence of some critical situations: illumination changes, camouflage, and shadows. To solve these problems, some authors have proposed to use other features like edge [31], texture [30], and stereo features [29], in addition to the color features. The stereo features deal with the camouflage but two cameras are needed. The edge feature allows us to deal with the local illumination changes and the ghost left when waking foreground objects begin to move [2]. The texture features are appropriate for both of illumination changes and shadows.

In the literature, fuzzy concepts have been used in foreground detection in two ways: The first interest [55] consists of avoiding a crisp decision when the comparison is made for the classification and the second one [3, 63] is to aggregate different features using fuzzy integrals. The idea is to be more robust to illumination changes and shadows.

5.3.1 Saturating Linear Function

In standard foreground detection, a crisp limiter function is used to classify pixels, as background or foreground:

$$F_t(x,y) = \begin{cases} 1, & \text{if } d\left(I_t(x,y), B_{t-1}(x,y)\right) > T \\ 0, & \text{otherwise,} \end{cases} \quad (5.20)$$

where $I_t(x,y)$, $B_{t-1}(x,y)$ and $F_t(x,y)$ are respectively the current image at time t, the background images at time $t-1$, and the foreground mask at time t. $B_{t-1}(x,y)$ can be obtained by any background modeling method. Using a running average as the background model, Sigari et al. [55] proposed the use of a saturating linear function instead of a crisp limiter as follows:

$$F_{f,t}(x,y) = \begin{cases} 1, & \text{if } d\left(I_t(x,y), B_{t-1}(x,y)\right) > T \\ \frac{|I_t(x,y) - B_{t-1}(x,y)|}{T}, & \text{otherwise,} \end{cases} \tag{5.21}$$

So, the result of fuzzy foreground detection is a real value in the range [0,1]. To obtain a binary foreground mask, Sigari et al. [55] applied a lowpass filter (LPF). In simple mode, a 3*3 mean filter is used. Then, the binary foreground mask is computed as follows:

$$F_t(x,y) = \begin{cases} 1, & \text{if } |LPF(F_{f,t}(x,y)| > T_f \\ 0, & \text{otherwise,} \end{cases} \tag{5.22}$$

where T_f is a threshold.

Sigari et al. [54] applied this fuzzy foreground detection in vehicle detection. The background model is a running average and the background maintenance is a fuzzy adaptive one (see Section 5.4.1). Results [54] show the pertinence of this approach in the case of camouflage.

Furthermore, Shakeri et al. [51, 52] have adapted this method in cellular automata for urban traffic applications. Each frame sequence is modeled by a cellular automata, and specific cellular automata rules are applied on pixels. Computation is done independently in all cells. Instead of using a trial and error procedure [55], this approach uses a specific threshold value that is obtained from the following equation:

$$F_t(x,y) = \begin{cases} 1, & \text{if } d\left(I_t(x,y), B_{t-1}(x,y)\right)/255 > exp(-(3+m)/2) * k \\ 0, & \text{otherwise,} \end{cases} \tag{5.23}$$

where $I_t(x,y)$, $B_{t-1}(x,y)$, and $F_t(x,y)$ are respectively the current image, the background images, and the foreground mask at time t. $B_{t-1}(x,y)$ can be obtained by any background modeling method. Note that dividing the distance between the background and current image assumes a number in the range [0,1]. In this equation, m is the number of steps that foreground detection procedure has been performed on this frame. Also, k is obtained by fuzzy sets.The initial values of m and k are respectively set to 0 and 1. Experimental results [52] show better performance for the fuzzy cellular foreground detection against the fuzzy foreground detection [55].

The background model used in Sigari et al. [55] and Shakeri et al. [52] is the running average and thus is unimodal. This fuzzy scheme can be used for multi-modal background as developed by Rossel-Ortega et al. [50]. The background model is a multi-modal such as in [48]. Foreground detection is made

by thresholding the background membership value. Experiments [50] show the performance on the Wallflower dataset against the crisp multi-modal [48] and the crisp unimodal [49] background model with confidence measure.

This fuzzy foreground detection based on a saturating linear function allows us to avoid crisp decisions and enhance the robustness to some critical situations such as camouflage in a traffic scene.

5.3.2 Fuzzy Integrals

Another way to perform the foreground detection is to use another feature with the color features and to aggregate the result by a fuzzy integral. As seen in the Section 5.1, the choice of the feature is essential and using more than one feature allows to be more robust to illumination changes and shadows. In this context, Zhang and Xu [63] have used texture and color features to compute similarity measures between current and background pixels. Then, these similarity measures are aggregated by applying the Sugeno integral. The assumption made by the authors reflects that the scale is ordinal. The moving objects are detected by thresholding the results of the Sugeno integral. Recently, El Baf et al. [3–5] have shown that the scale is continuum in the foreground detection. Therefore, they used the same features with the Choquet integral instead of the Sugeno integral. Ding et al. [22] used the Choquet integral too but they change the similarity measures. Azab et al. [1] have aggregated recently three features, i.e color, edge and texture. The Figure 5.6 shows the fuzzy foreground detection process with the color and texture features. The fuzzy integral can be the Sugeno integral [63] or the Choquet

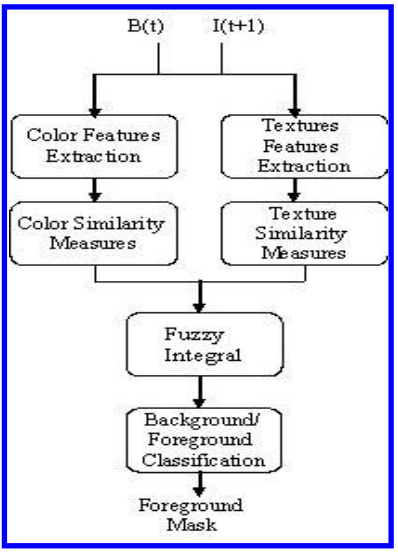


FIGURE 5.6 Fuzzy foreground detection process.

integral [3]. We describe in the following each step of this process with the Choquet integral. Furthermore, the process is similar for more than two features.

Color and textures features

- *Color Features:* The selection of the color space, such as color features, is one of the key factors for efficient color information extraction. In foreground detection, the most commonly used is the RGB space, because it is the one directly available from the sensor or the camera. The RGB color space has an important drawback: their three components are dependent which increase 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. A number of color space comparisons are presented in the literature [32,35,47] and usually YCrCb is selected as the most appropriate color space. But first, let us define the different color spaces (the Ohta, HSV, and YCrCb) tested that separate the luminance and the chrominance channels. The axes of the Ohta space [45] are the three largest eigenvectors of RGB space, found from the principal components analysis of a large selection of natural images. This color space is a linear transformation of RGB and has three components I_1,I_2, and I_3. Equation (5.24) shows the relationship between RGB to the Ohta space:

$$\begin{aligned} I_1 &= (R+G+B)/3 \\ I_2 &= (R-B)/2 \ \ (or \ \ (B-R)/2) \\ I_3 &= (2G-R-B)/4 \end{aligned} \tag{5.24}$$

HSV and YCrCb are closer to human interpretation of colors in the sense that brightness, or intensity, is separated from the base color. YCrCb uses cartesian coordinates to describe the base color while HSV uses polar coordinates. For HSV, the color information improves the discrimination between shadow and object, classifying as shadows those pixels having approximately the same hue and saturation values compared to the background, but lower luminosity. Equations (5.25) and (5.26) below show the relationships between RGB and HSV, then YCrCb color spaces:

$$\begin{aligned} H &= \begin{cases} 60\,(G-B)/\Delta & \text{if } max\,(R,G,B)=R \\ 60\,(B-R)/(\Delta+120) & \text{if } max\,(R,G,B)=G \\ 60\,(R-G)/(\Delta+240) & \text{if } max\,(R,G,B)=B \end{cases} \\ S &= \Delta/max\,(R,G,B) \\ V &= max\,(R,G,B)\,, \end{aligned} \tag{5.25}$$

where $\Delta = max\,(R,G,B) - min\,(R,G,B)$.

$$\begin{aligned} Y &= 0.25R + 0.504G + 0.098B + 16 \\ Cr &= 0.439R - 0.368G - 0.071B + 128 \\ Cb &= -0.148R - 0.291G + 0.439B + 128 \end{aligned} \tag{5.26}$$

For each color space, two components are chosen according to their sensitivity to illumination changes. For example, the components Cr and Cb are retained as the luminance Y is very sensitive to illumination changes.

- *Texture Feature:* The texture feature used is the Local Binary Pattern (LBP), which was developed by Heikkila and Pietikinen [30]. The LBP is invariant to monotonic changes in gray scale, which makes it robust against illumination changes. The operator labels the pixels of an image block by thresholding the neighborhood of each pixel with the centre value and considering the result as a binary number:

$$LBP\,(x,y) = \sum_{i=0}^{N-1} s\,(g_i - g)\,2^i,$$

where g corresponds to the gray value of the center pixel (x, y) and g_i to the gray values of the N neighborhood pixels. The function s is defined as follows:

$$s\,(z) = \begin{cases} 1 & if \quad z \geq 0 \\ 0 & if \quad z < 0. \end{cases}$$

The original LBP operator worked with the 3×3 neighbourhood of a pixel.

Once the features are chosen, we can compare the background and current images using similarity measures that have to be defined.

Similarity measures

The comparison is made using a similarity measure between pixels in the current and background images. We define in the following the similarity measures for the color and the texture features as they were developed in [5] but other similarity measures can be found in [1, 22].

- *Color Similarity Measures:* We describe the similarity measure in a general way, that is, the color features may be any color space with three components noted I_1, I_2 and I_3. Then, the color similarity measure $S_k^C\,(x,y)$ at the pixel (x, y) is computed as in [63]:

$$S_k^C\,(x,y) = \begin{cases} \frac{I_k^C(x,y)}{I_k^B(x,y)} & if \quad I_k^C\,(x,y) < I_k^B\,(x,y) \\ 1 & if \quad I_k^C\,(x,y) = I_k^B\,(x,y) \\ \frac{I_k^B(x,y)}{I_k^C(x,y)} & if \quad I_k^C\,(x,y) > I_k^B\,(x,y)\,, \end{cases}$$

where $k \in \{1, 2, 3\}$ is one of the three color features. B and C represent respectively the background and the current images at time t. B can be obtained using any of the background modelling method. Note that $S_k^C(x, y)$ is between 0 and 1. Furthermore, $S_k^C(x, y)$ is closed to 1 if $I_k^C(x, y)$ and $I_k^B(x, y)$ are very similar.

- *Texture Similarity Measure:* The texture similarity measure $S^T(x, y)$ at the pixel (x, y) is computed as follows:

$$S^T(x, y) = \begin{cases} \frac{L^C(x,y)}{L^B(x,y)} & if \quad L^C(x, y) < L^B(x, y) \\ 1 & if \quad L^C(x, y) = L^B(x, y) \\ \frac{L^B(x,y)}{L^C(x,y)} & if \quad L^C(x, y) > L^B(x, y), \end{cases}$$

where $L^B(x, y)$ and $L^C(x, y)$ are respectively the texture LBP of pixel (x, y) in the background and current images at time t. Note that $S^T(x, y)$ is between 0 and 1. Furthermore, $S^T(x, y)$ is close to 1 if $L^B(x, y)$ and $L^C(x, y)$ are very similar.

Once these two similarity measures are computed, they can be aggregated by a fuzzy integral. We remind in the following some concepts on fuzzy integrals and then we explain the aggregation of the similarity measures with the Choquet integral.

Fuzzy integrals: A brief background

There are several mathematical operators used for the aggregation of different measures. In the literature [21], we find the basic ones like the average, the median, the minimum, and the maximum, as well as some generalizations like the Ordered Weighted Average (OWA) having the minimum, and the maximum as particular cases and k-order statistics. Then, the family of fuzzy integrals has presented through its discrete version a generalization of OWA or the weighted average using the Choquet integral, as well as the minimum and the maximum using the Sugeno integral. The advantage of fuzzy integrals is that they take into account the importance of the coalition of any subset of criteria. We briefly summarize the necessary concepts around fuzzy integrals (Sugeno and Choquet).
Let μ be a fuzzy measure on a finite set X of criteria and $h : X \to [0, 1]$ be a fuzzy subset of X.

Definition 1 *The Sugeno integral of h with respect to μ is defined by*

$$S_\mu = Max\left(Min\left(h\left(x_{\sigma(i)}\right), \mu\left(A_{\sigma(i)}\right)\right)\right), \tag{5.27}$$

where σ is a permutation of the indices such that
$h\left(x_{\sigma(1)}\right) \leq \ldots \leq h\left(x_{\sigma(n)}\right)$ *and* $A_{\sigma(i)} = \{\sigma(1), \ldots, \sigma(n)\}$.

Definition 2 *The Choquet integral of h with respect to μ is defined by*

$$C_\mu = \sum_{i=0}^{n} h\left(x_{\sigma(i)}\right)\left(\mu\left(A_{\sigma(i)}\right) - \mu\left(A_{\sigma(i+1)}\right)\right) \tag{5.28}$$

with the same notations as above.

An interesting interpretation of the fuzzy integral arises in the context of the source fusion. The measure μ can be viewed as the factor that describes the relevance of the sources of information, where h denotes the values that the criteria have reported. The fuzzy integrals then aggregate nonlinearly the outcomes of all criteria. The Choquet integral is adapted for cardinal aggregation while the Sugeno integral is more suitable for ordinal aggregation. More details can be found in [44, 58, 59].

In the fusion of different criteria or sources, the fuzzy measures take on an interesting interpretation. A pixel can be evaluated based on criteria or sources providing information about the state of the pixel, whether the pixel corresponds to background or foreground. The more the criteria provide information about the pixel, the more relevant the decision of the pixel's state. Let $X = \{x_1, x_2, x_3\}$, with each criterion, we associate a fuzzy measure $\mu(x_1) = \mu(\{x_1\})$, $\mu(x_2) = \mu(\{x_2\})$ and $\mu(x_3) = \mu(\{x_3\})$ such that the higher the $\mu(x_i)$, the more important the corresponding criterion in the decision. To compute the fuzzy measure of the union of any two disjoint sets whose fuzzy measures are given, we use an operational version proposed by Sugeno that is called the λ-*fuzzy measure*. To avoid excessive notation, let us denote this measure by μ_λ-*fuzzy measure*, where λ is a paramater of the fuzzy measure used to describe an interaction between the criteria that are combined. Its value can be determined through the boundary condition, that is, $\mu(X) = \mu(\{x_1, x_2, x_3\}) = 1$. The fuzzy density values over a given set $K \subset X$ is computed as

$$\mu_\lambda(K) = \frac{1}{\lambda}\left[\prod_{x_i \in K}\left(1 + \lambda\mu_\lambda(x_i)\right) - 1\right]. \tag{5.29}$$

Aggregation of color and texture similarity measures by a fuzzy integral

As defined above, the computed measures are obtained by dividing the intensity values in the background and current images by endpoints denoted by 0 and 1, where 0 means that the pixels at the same location in background and current images respectively are not similar, and 1 means that these pixels are similar, that is, the pixel corresponding to background. In such a case, the scale is a continuum and is constructed as a cardinal one where the distances or the differences between values can be defined. For example, the distance between 0.1 and 0.2 is the same as the distance between 0.8 and 0.9, because

numbers have a real meaning. In the case of an ordinal scale, the numbers correspond to modalities when an order relation on the scale should be defined. A typical example of the former is when we define a scale [a, b, c, d, e] to evaluate the level of some students, where "a" corresponds to "excellent" and "e" to "very bad." So that, the difference between "b" (very good) and "c" (good) is not necessarily the same as the difference between "c" (good) and "d" (bad). Hence, operations other than comparison on a cardinal scale can be allowed such as standard arithmetic operations, typically addition and multiplication. In this sense, the Choquet integral is considered more suitable than the Sugeno integral because of its ability to aggregate well the features on a cardinal scale and to use such arithmetic operations. So, for each pixel, color and texture similarity measures are computed, as explained previously from the background and the current frame. We define the set of criteria $X = \{x_1, x_2, x_3\}$ with, (x_1, x_2) = two components color features of the chosen color space (i.e. Ohta, HSV, YCrCb, etc..) and x_3 = texture feature obtained by the LBP.

For each x_i, let $\mu(x_i)$ be the degree of importance of the feature x_i in the decision whether the pixel corresponds to background or foreground. The fuzzy functions $h(x_i)$ are defined in $[0, 1]$ so that, $h(x_1) = S_1^C(x, y)$, $h(x_2) = S_2^C(x, y)$ and $h(x_3) = S^T(x, y)$. To compute the value of the Choquet integral for each pixel, we need first to rearrange the features x_i in the set X with respect to the order: $h(x_1) \geq h(x_2) \geq h(x_2)$.

The pixel at position (x, y) is considered foreground if its Choquet integral value is less than a certain constant threshold Th:

$$if \quad C_\mu(x, y) < Th \quad then \quad (x, y) \ is \ foreground,$$

which means that pixels at the same position in the background and the current images are not similar. Th is a constant value depending on each video dataset.

Results presented in Section 5.5.2 show the relevance of the fuzzy integrals in the presence of illuminations changes and shadows.

5.4 Background Maintenance

The background maintenance determines how the background will adapt itself to take into account the critical situations that can occur. In the literature, there are two maintenance schemes: the blind one and the selective one.

Blind background maintenance consists of updating all the pixels with the same rules, which is usually an IIR filter, as follows:

$$B_t(x, y) = (1 - \alpha) B_{t-1}(x, y) + \alpha I_t(x, y), \tag{5.30}$$

where α is a learning rate that is a constant in the interval $[0, 1]$.

The disadvantage of this scheme is that the value of pixels classified as foreground are taken into account in the computation of the new background

and thus pollute the background image. To solve this problem, some authors use a selective maintenance which consists of computing the new background image with a different learning rate following its previous classification into foreground or background as follows:

$$\begin{aligned} B_t(x,y) &= (1-\alpha) B_{t-1}(x,y) + \alpha I_t(x,y) \\ &\quad if\ (x,y)\ is\ background \\ B_t(x,y) &= (1-\beta) B_{t-1}(x,y) + \beta I_t(x,y) \\ &\quad if\ (x,y)\ is\ foreground. \end{aligned} \tag{5.31}$$

Here, the idea is to adapt very quickly a pixel classified as background and very slowly a pixel classified as foreground. For this reason, $\beta << \alpha$ and usually $\beta = 0$. So the second expression in Equation (5.31) becomes

$$B_t(x,y) = B_{t-1}(x,y). \tag{5.32}$$

In this crisp decision, there are two learning rates: one learning rate for pixels classified as background and another for pixels classified as foreground. This crisp decision generates the fact that erroneous classification results may make a permanent incorrect background model. To avoid this problem, an adaptive learning rate that integrates the membership of the pixels to one of the class background or foreground seems to be better. So, the adaptive background maintenance consists of the following:

$$B_t(x,y) = (1-\alpha_t(x,y)) B_{t-1}(x,y) + \alpha_t(x,y) I_t(x,y), \tag{5.33}$$

where $\alpha_t(x,y)$ is an adaptive learning rate that is a variant in the interval $[0,1]$ and depends on the membership of the pixels in the class background or foreground. Different approaches were proposed in the literature to compute $\alpha_t(x,y)$.

5.4.1 Fuzzy Learning Rates

Sigari el al. [54, 55] proposed to compute an adaptive learning rate $\alpha_t(x,y)$ at each pixel following the fuzzy membership value obtained by each pixel in fuzzy foreground detection (see Section 5.3.1). Therefore, $\alpha_t(x,y)$ is computed as follows:

$$\alpha_t(x,y) = (1-\alpha_{min}) exp(-5 * F_{f,t}(x,y)), \tag{5.34}$$

where $F_{f,t}(x,y)$ is the fuzzy foreground mask obtained by the saturating linear function (see Section 5.3.1) and α_{min} is the minimum value for α. In Sigari et al. [54, 55], α_{min} is set to 0.9.

In the same idea to adapt the background, Maddalena and Petrosino [39]

proposed an adaptive learning rate in a neural network background modeling. In this case, $\alpha_t(x,y)$ is computed as follows:

$$\alpha_t(x,y) = (1 - F_t(x,y)) * \alpha_t * w_{x,y}, \tag{5.35}$$

where $w_{x,y}$ are Gaussians weights in the n*n neighborhood, α_t represents the learning factor that is the same for each pixel of the t-th sequence frame and depends on the scene variability, and $F_t(x,y)$ is a saturating linear function given by:

$$F_t(x,y) = \begin{cases} \frac{d(c_m(x,y),p_t(x,y))}{\epsilon}, & \text{if } d(c_m(x,y),p_t(x,y)) < \epsilon \\ 1, & \text{otherwise,} \end{cases} \tag{5.36}$$

where $c_m(x,y)$ is a weight vector coming from the neural network and corresponds to the background value at the pixel (x,y). $p_t(x,y)$ is the current value of the pixel (x,y), and ϵ is a constant threshold. $F_t(x,y)$ can be interpreted as the membership value of the pixel (x,y) to the background model. So, the closer the current pixel (x,y) is to the background model, the more it can contribute to the maintenance.

Recently, Maddalena and Petrosino [40,41] performed the adaptivity by introducing spatial coherence information. In this scheme, $\alpha_t(x,y)$ is computed as follows:

$$\alpha_t(x,y) = (1 - F_{1,t}(x,y) * F_{2,t}(x,y)) * \alpha_t * w_{x,y}, \tag{5.37}$$

where the function $F_{1,t}$ is the adaptive fuzzy factor and $F_{2,t}$ is the fuzzy coherence factor. So, $F_{1,t}(x,y)$ is equal to $F_t(x,y)$, which is defined in Equation (5.36). $F_{2,t}$ is defined as follows:

$$F_{2,t}(x,y) = \begin{cases} 2NCF(p_t(x,y)) - 1, & \text{if } NCF(p_t(x,y)) > 0.5 \\ 1, & \text{otherwise,} \end{cases} \tag{5.38}$$

where $NCF(p_t(x,y))$ is a neighborhood coherence factor defined in [40]. This factor is based on the fact that the greater $NCF(p_t(x,y))$ is, the greater is the number of the pixels in the neigborhood of the pixel (x,y) that represented the background model and then the better the pixel (x,y) represents the background.

Comparative results between the adaptive learning rate [55] and the adaptive learning rate with spatial coherence [40] are presented qualitatively in Section 5.5.3 and quantitatively in Section 5.5.4.

5.4.2 Fuzzy Maintenance Rules

One disadvantage of selective maintenance is mainly due to the crisp decision that attributes a different rule following the classification in background or foreground. To solve this problem, El Baf et al. [8] proposed to take into

account the uncertainty of the classification. This can be made by graduating the update rule using the result of the Choquet integral as follows:

$$B_t(x,y) = \mu_F B_{t-1} + (1-\mu_F)\left((1-\alpha)B_{t-1}(x,y) + \alpha I_t(x,y)\right), \quad (5.39)$$

where $\mu_F = 1 - \mu_B$. μ_F and μ_B are respectively the fuzzy membership values of the pixel (x,y) to the class foreground and background. μ_B is a function of $C_\mu(x,y)$ such as $\mu_B = 1$ for $Max(C_\mu(x,y))$ and $\mu_B = 0$ for $Min(C_\mu(x,y))$.

We can note that adaptive maintenance is a generalized version of selective maintenance. Indeed, if the pixel is classified as background with the Choquet integral value equal to 1, we retrieve the first expression of Equation (5.31), and if the pixel is classified as foreground with the Choquet integral value equal to 0, Equation (5.40) is equal to Equation (5.32). This fuzzy adaptive scheme is tested on the Wallflower test in Section 5.5.3.

5.5 Experimental Results

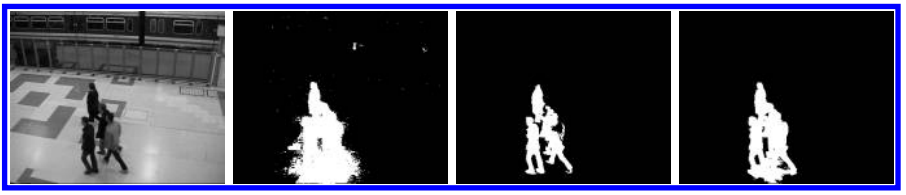
We have evaluated the different existing fuzzy approaches. First, we compare the fuzzy method with their corresponding crisp method. Then, we evaluate them on the same dataset [53].

5.5.1 Fuzzy Background Modeling

In this section, the experiments are conducted to compare the results of T2-FGMM [6] with the crisp GMM [57]. So, we have applied the GMM, T2-FGMM-UM, and T2-FGMM-UV algorithms to indoor and outdoor videos where different critical situations occur, such as camera jitter, movement in the background, illuminations changes, and shadows. For T2-FGMM, the best results were obtained with the values 2 and 0.9, respectively, for the factors k_m and k_ν.

Indoor scene videos

The PETS 2006 dataset [27] provides several video presenting indoor sequences in a video surveillance context. In these video sequences, there are illumination changes and shadows. Figure 5.7 presents the results obtained by GMM [57], T2-FGMM-UM, and T2-FGMM-UV. It is noticed that the results obtained using T2-FGMM-UM and T2-FGMM-UV are better than using the crisp GMM. The silhouettes are well detected with the T2-FGMM-UM. T2-FGMM-UV is more sensitive because the variance is more unstable over time.



a) Current Image b) GMM c) T2-FGMM-UM d) T2-FGMM-UV

FIGURE 5.7 Background subtraction with illumination changes (PETS 2006 dataset). From left to right: a) The original image (Frame 165), b) the mask obtained by the GMM, c) the mask obtained using T2-FGMM-UM, d) the mask obtained using T2-FGMM-UV [6]. (©2008 Springer Verlag)

Outdoor scene videos

We have chosen three videos presenting different dynamic backgrounds: camera jitter, waving trees, and water rippling. The first outdoor sequence that was tested involved a camera mounted on a tall tripod and comes from [37]. The wind caused the tripod to sway back and forth, causing nominal motion in the scene. In Figure 5.8, the first row shows different current images. The second row shows the results obtained by the GMM proposed in [57]. It is evident that the motion causes substantial degradation in performance. The third and fourth rows show respectively the results obtained by T2-FGMM-UM and T2-FGMM-UV. As for the indoor scene, T2-FGMM-UM and T2-FGMM-UV give better results than the crisp GMM.

We have also tested our method for the sequences Campus and Water Surface that come from [36]. Figure 5.9 shows the robustness of T2-FGMM-UM against waving trees and water rippling.

5.5.2 Fuzzy Foreground Detection

We have applied the Sugeno and Choquet integrals to different datasets: the first one is the PETS dataset. We have chosen particularly the PETS 2000 [26] and PETS 2006 [27] datasets used in video surveillance. The output images of these two datasets are respectively 768×576 and 720×576 pixels. The second one is our Aqu@theque dataset used in a multimedia application [2], where the output images are 384×288 pixels. The results are obtained without post processing and the threshold for each algorithm is optimized to give the best results.

PETS 2000 and 2006 dataset

We first tested the Sugeno and Choquet integrals on the PETS 2000 and 2006 benchmark data indoor and outdoor sequences in video surveillance context. The goal is to detect moving persons and/or vehicules. The use of the Choquet integral [3] with YCrCb color space shows more robustness to illumination

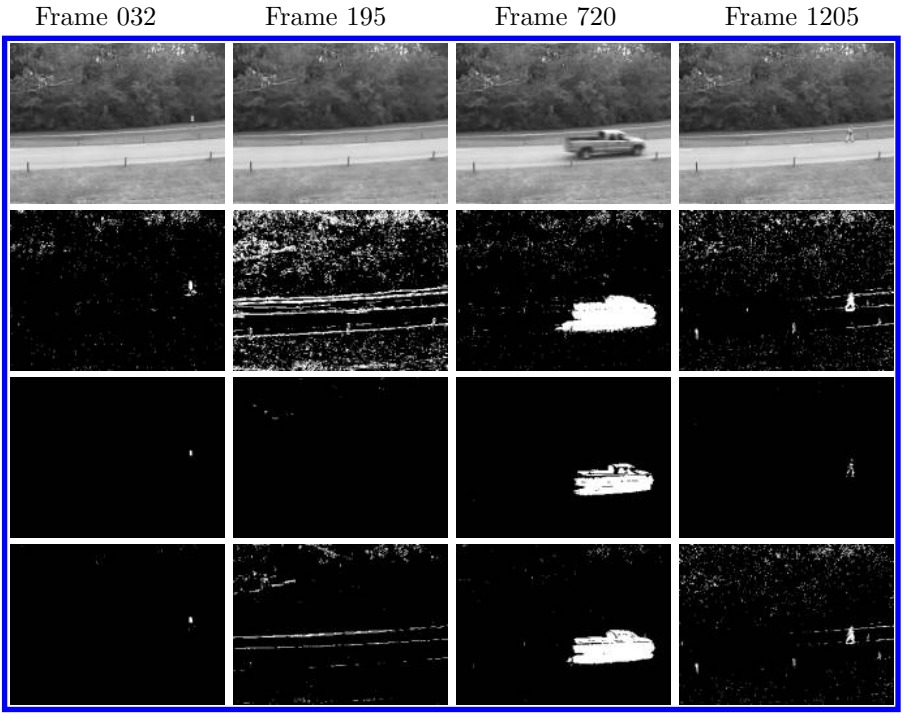


FIGURE 5.8 Fuzzy background modeling: The first row shows the original frames for the sequence Camera Jitter. The second row presents the segmented images obtained by GMM. The third and fourth rows illustrate the result obtained using T2-FGMM-UM and T2-FGMM-UV, respectively [6]. (©2008 Springer Verlag.)

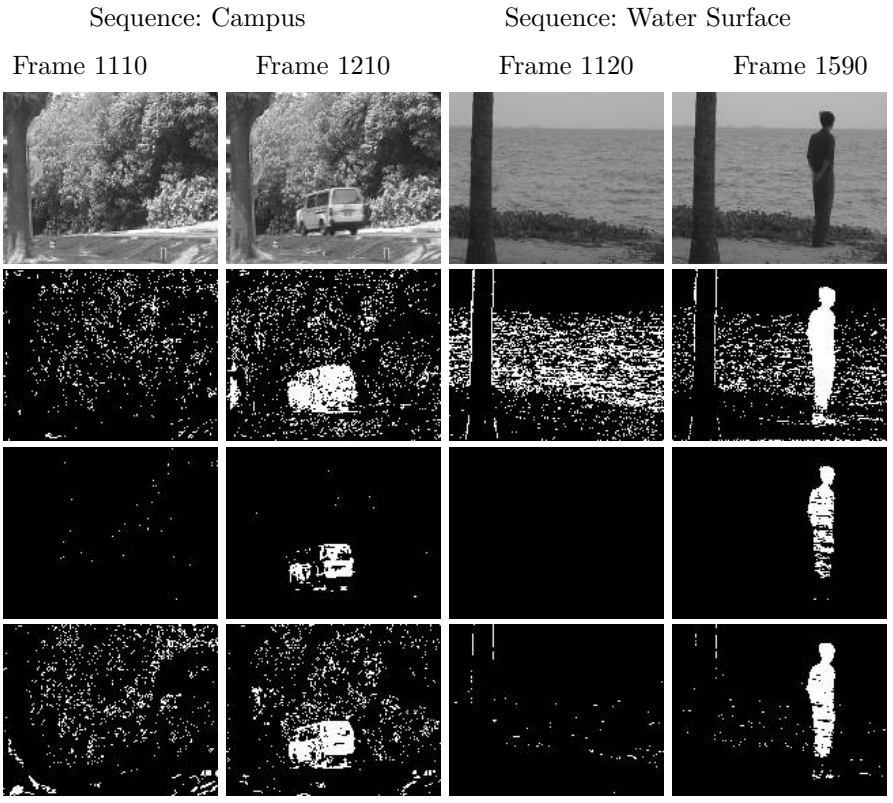
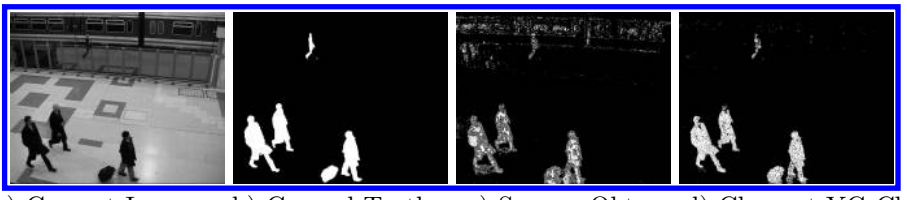


FIGURE 5.9 Fuzzy background modeling: The first row shows the original frames for Campus and Water Surface sequences. The second row presents the segmented images obtained by GMM. The third and fourth rows illustrate the result obtained using T2-FGMM-UM and T2-FGMM-UV, respectively [6]. (©2008 Springer Verlag.)

changes and shadows than the Sugeno integral [63], as we can see qualitatively in the Figures 5.10 and 5.11.

a) Current Image b) Sugeno-Ohta c) Choquet-YCrCb

FIGURE 5.10 From left to right: a) The current image, b) the mask obtained by the Sugeno integral using the Ohta space, c) the mask obtained by the Choquet integral using the YCrCb space [5]. (©2008 IEEE.)



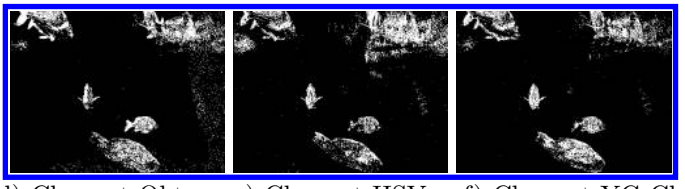
a) Current Image b) Ground Truth c) Sugeno-Ohta d) Choquet-YCrCb

FIGURE 5.11 From left to right: a) The current image, b) the ground truth, c) the mask obtained by the Sugeno integral using the Ohta space, d) the mask obtained by the Choquet integral using the YCrCb space [5]. (©2008 IEEE.)

Aqu@theque dataset

This dataset contains several video sequences presenting fish in a tank. The goal of the application Aqu@theque [2] is to detect fish and identify them. In these aquatic video sequences, there are many critical situations. For example, there are illumination changes due to the ambient light, the spotlights that light the tank from the inside and from the outside, the movement of the water due to fish, and the continuous renewal of the water. These illumination changes can be local or global, following their origin. Furthermore, the constitution of the aquarium (rocks, algae) and the texture of the fish amplify the consequences of the brilliant variation. Figure 5.12 shows the experiments made on one sequence. In Table 5.2, we show the fuzzy density values that we have tested. The best results are obtained with $\{0.53, 0.0.34, 0.13\}$. It is noticed that the results obtained using the Choquet integral [3] are better than those using the Sugeno integral [63] with the same color space, that is, Ohta. The results obtained with the Choquet integral using other color spaces, that

a) Current Image b) Ground Truth c) Sugeno-Ohta



d) Choquet-Ohta e) Choquet-HSV f) Choquet-YCrCb

FIGURE 5.12 First row: The current image, the ground truth, Sugeno-Ohta. Second row: Choquet-Ohta, Choquet-HSV, and Choquet-YCrCb [5]. (©2008 IEEE.)

is, the HSV and YCrCb, confirmed that optimum results are obtained using the Choquet integral with the YCrCb color features.

TABLE 5.2 Fuzzy measure values.

$\{x_1\}$	$\{x_2\}$	$\{x_3\}$	$\{x_1, x_2\}$	$\{x_1, x_3\}$	$\{x_1, x_3\}$	$\{X\}$
0.6	0.3	0.1	0.9	0.7	0.4	1
0.5	0.4	0.1	0.9	0.6	0.5	1
0.5	0.3	0.2	0.8	0.7	0.5	1
0.50	0.39	0.11	0.89	0.61	0.5	1
0.53	0.34	0.13	0.87	0.66	0.47	1

The quantitative evaluation was done first using the similarity measure derived by Li [37]. Let A be a detected region and B be the corresponding ground truth, the similarity between A and B can be defined as

$$S(A, B) = \frac{A \cap B}{A \cup B}. \tag{5.40}$$

If A and B are the same, $S(A, B)$ approaches 1, otherwise 0; that is, A and B have the least similarity. The ground truth is marked manually. Table 5.3 shows the similarity value obtained for the previous experiments. It is

TABLE 5.3 Similarity measure.

Integral Color Space	Sugeno Ohta	Choquet Ohta	Choquet HSV	Choquet YCrCb
$S(A, B)$	0.27	0.44	0.34	0.46

well identified that optimum results are obtained by the Choquet integral.

Furthermore, the Ohta and the YCrCb spaces give almost similar results ($S_{Ohta} = 0.44, S_{YCrCb} = 0.46$), when the HSV space registers ($S_{HSV} = 0.34$). When observing the effect of YCrCb and Ohta spaces on the images, we have noticed that the YCrCb is slightly better than the Ohta space. To see the progression of the performance of each algorithm, we use the ROC curves [20]. For that, we compute the false positive rate (FPR) and the true positive rate (TPR) as follows:

$$FPR = \frac{FP}{FP + TN}, \qquad TPR = \frac{TP}{TP + FN},$$

where TP is the total of true positives, TN is the total of true negatives, FP is the total of false positives, and FN is the total of false negatives. The FPR is the proportion of background pixels that was erroneously reported as being moving object pixels. And the TPR is the proportion of moving object pixels that was correctly classified among all positive samples. Figure 5.13 represents the ROC curves for the Sugeno and the Choquet integrals with the Ohta color space. These curves confirm that the Choquet integral outperforms the Sugeno one using the Ohta space.

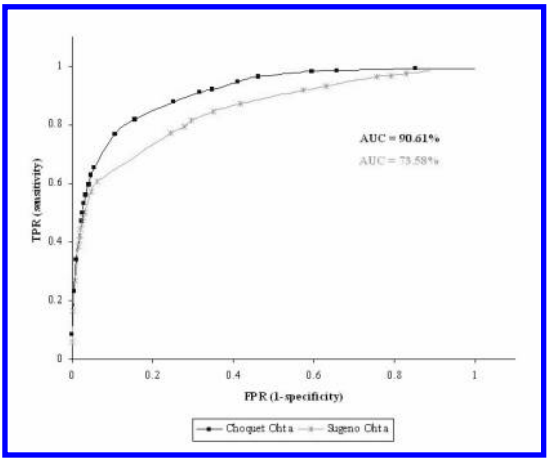


FIGURE 5.13 ROC curve: Comparison of the two detection algorithms using respectively the Sugeno and the Choquet integrals in the Ohta color space [5].(©2008 IEEE.)

Then, we compared the previous results with other color spaces. Figure 5.14 shows the ROC curves for the Choquet integral with the Ohta, HSV, and YCrCb color spaces. Once again, the curves confirm the previous conclusion. Indeed, the Areas Under Curve (AUCs) are very similar for YCrCb and Ohta spaces.

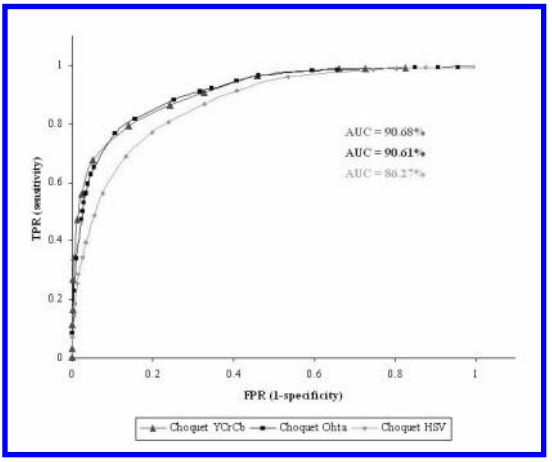


FIGURE 5.14 ROC curve: Evaluation of the effect of different color spaces on the detection algorithm using the Choquet integral [5].(©2008 IEEE.)

5.5.3 Fuzzy Background Maintenance

This section presents tests of the fuzzy learning rates and the fuzzy maintenance rules on the Wallflower dataset [60], which contains seven real video sequences. Each one of them presents typical critical situations.

Fuzzy learning rates

This section presents the results obtained by the fuzzy learning rates from [55] and the fuzzy learning rate with spatial coherence from [40, 41] on the sequence called "Waving Trees". Figure 5.15 shows the original image, the ground truth, the foreground mask obtained by [55] and the foreground mask obtained by [40]. The optimum results are obtained by the fuzzy learning rate with spatial coherence.

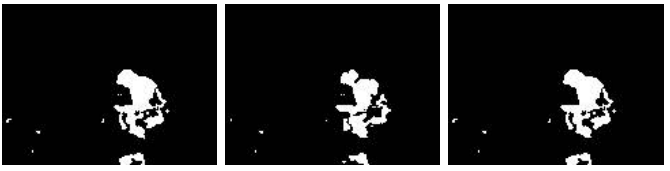
a) Current Image b) Ground Truth c) FBGS [55] d) SOBS-CF [40]

FIGURE 5.15 From left to right: a) The original image, b) the ground truth, c) the foreground mask obtained by [55] and d) the foreground mask obtained by [40,41]. (©2010 Springer Verlag.)

Fuzzy maintenance rules

We have tested the fuzzy maintenance rules [8] on the sequence called "Time of Day". We choose this sequence because it presents gradual illumination changes that alter the background and thus permit us to show the performance of the fuzzy maintenance rules. Figure 5.16 shows the original image, the ground truth, the final foreground mask obtained by the blind scheme, the selective one, and the fuzzy one. Table 5.4 shows the similarity value obtained

a) Current Image b) Ground Truth



c) Blind d) Selective e) Fuzzy Adaptive

FIGURE 5.16 First row: a) the current image, b) the ground truth. Second row: c) blind maintenance, d) selective maintenance, e) fuzzy adaptive maintenance [8]. (©2008 IEEE.)

for the previous experiments. The optimum results are obtained by the fuzzy adaptive scheme.

TABLE 5.4 Similarity measure.

Maintenance Scheme	Blind Maintenance	Selective Maintenance	Adaptive Maintenance
$S(A, B)$ %	58.40	57.08	58.96

5.5.4 Comparison

In this section, we compare, on the same dataset, the following fuzzy approaches: background modeling by Type-2 Fuzzy GMMs [6], foreground detection by a linear saturation function [55], foreground detection by the Sugeno integral [63], foreground detection by the Choquet integral [3], and the background maintenance with the fuzzy spatial-coherence learning rate [41]. All the methods are compared in their original version and their parameters have been adapted until the results seem optimal over the entire sequence:

- ***Type-2 Fuzzy Gaussian Mixture Models (Section 5.2):*** The background initialization was made on 100 frames. The factors k_m and k_ν were set respectively to the values 2 and 0.9.
- ***Linear Saturation Function (Section 5.3.1):*** The background model is based on the running average. The method is called FBGS as in [41].
- ***Sugeno Integral and Choquet Integral (Section 5.3.2):*** The background model used is the running average applied on 100 frames. For the Sugeno integral, the color space is the Ohta space, while for the Choquet integral it is the YCrCb. The fuzzy measures values are the ones indicated in Table 5.2.
- ***Fuzzy spatial-coherence learning rate(Section 5.4.1):*** The background model is a self-organization through the artificial neural network. This method is called SOBS-CF as in [41].

This dataset comes from [53] and consists of a sequence of 500 frames of 360*240 pixels with ground truth masks. The camera is mounted on a tall tripod and the wind caused it to sway back and forth, causing nominal motion in the scene. This scene consists of a street crossing, where several people and cars pass by. Figure 5.18 shows the original test images in the first row and the corresponding ground truth in the second row while the corresponding results obtained by the fuzzy approaches, that is, T2-FGMM-UM, T2-FGMM-UV, Sugeno, Choquet, FBGS, and SOBS-CF, are reported in this order in the following rows.

We used ground truth-based metrics computed from the true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) as defined in Section 5.5.2. Table 5.5 shows the FP and FN for the different fuzzy approaches on the frames 271, 373, 410, and 465.

TABLE 5.5 Performance analysis.

Method	Error Type	Frame 271	Frame 373	Frame 410	Frame 465	Total Error
GMM	FN	0	1120	4818	2050	18576
	FP	2093	4124	2782	1589	
T2-FGMM-UM	FN	0	1414	6043	2520	10631
	FP	203	153	252	46	
T2-FGMM-UV	FN	0	957	2217	1069	10670
	FP	3069	1081	1119	1158	
Sugeno Ohta-LBP	FN	0	1852	7370	3210	13213
	FP	210	146	203	222	
Choquet YCrCb-LBP	FN	0	1243	5871	2592	9995
	FP	54	58	137	40	
FBGS	FN	0	431	814	249	4900
	FP	2034	288	400	684	
SOBS-CF	FN	0	147	298	91	1132
	FP	0	125	292	179	

Then, we computed the following metrics: detection rate,precision and F-measure. Detection rate gives the percentage of corrected pixels classified as

background when compared with the total number of background pixels in the ground truth:

$$DR = \frac{TP}{TP + FN}. \tag{5.41}$$

Precision gives the percentage of corrected pixels classified as background as compared to the total pixels classified as background by the method:

$$Precision = \frac{TP}{TP + FP}. \tag{5.42}$$

A good performance is obtained when the detection rate is high without altering the precision. We also computed the F-measure used in [41] as follows:

$$F = \frac{2 * DR * Precision}{DR + Precision}. \tag{5.43}$$

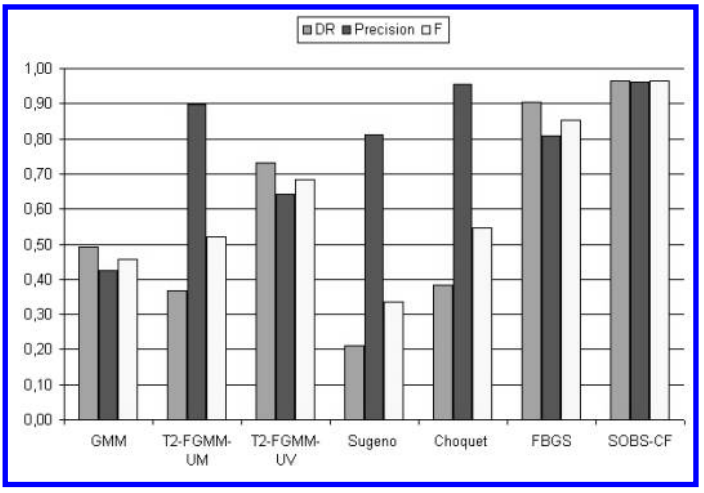


FIGURE 5.17 Overall performance.

Figure 5.17 shows the detection rate, precision and F-measure obtained for each fuzzy approach. We can observe that T2-FGMM-UM and T2-FGMM-UV outperform the crisp GMM. Furthermore, the Choquet integral shows better performance than the Sugeno integral. It confirms that the Choquet integral is more appropriate for foreground detection. Finally, the SOBS-CF algorithm performs better than all other methods due to the background model and the spatial information taken in the adaptive learning rate.

5.6 Conclusion

This chapter attempted to provide a comprehensive survey of research on fuzzy background subtraction. Thus, we reviewed the contribution of the fuzzy

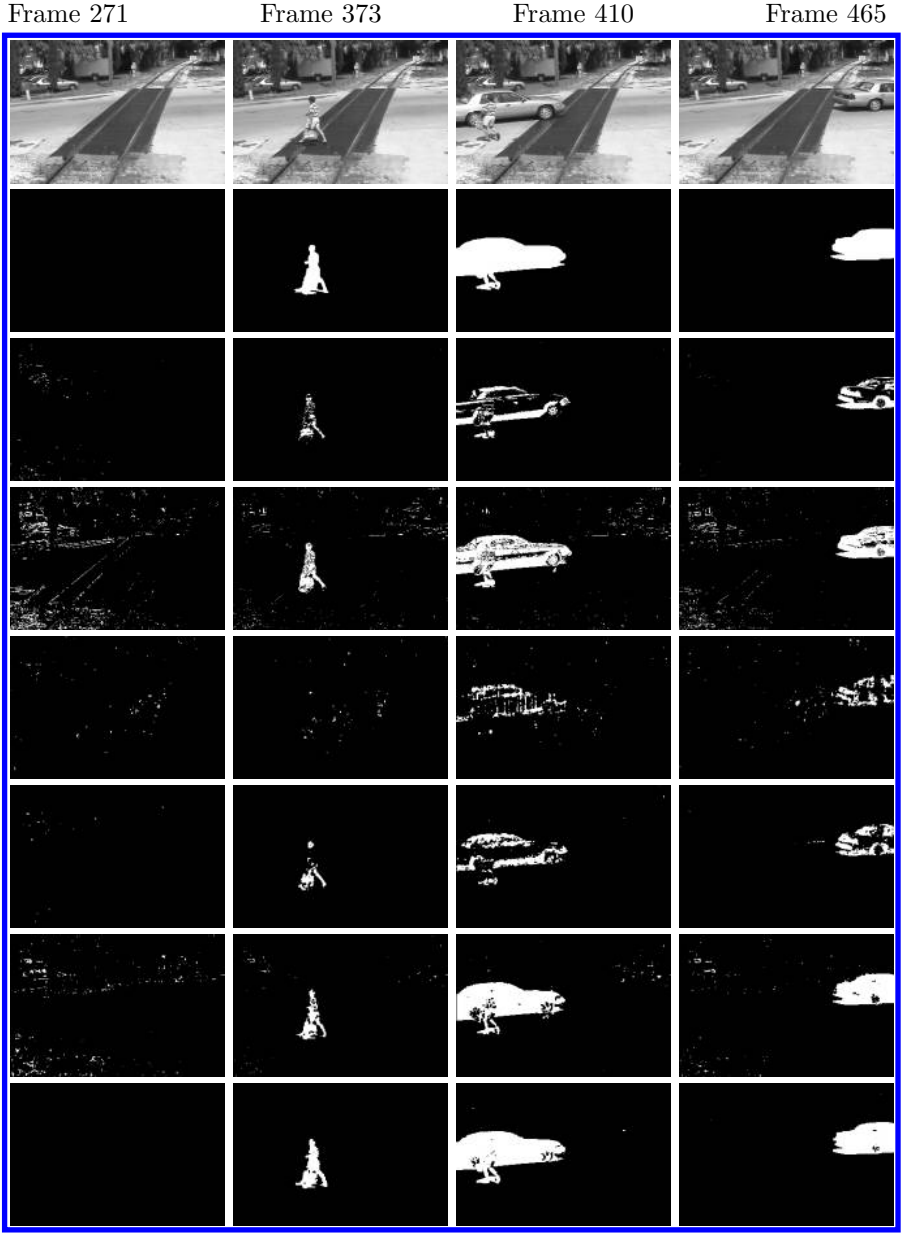


FIGURE 5.18 The first row contains the original images, the second row contains the ground truth images, the third row contains the results obtained using T2-FGMM-UM, and the fourth contains the results obtained by using T2-FGMM-UV, the fifth row contains the results obtained by using FBGS, the sixth and the seventh rows contain respectively the results obtained by using the Sugeno and the Choquet integrals, and the last row contains the results obtained by SOBS-CF. Images from [6] (©2008 Springer Verlag) and [41] (©2010 Springer Verlag).

concepts in each step. The Type-2 Fuzzy Gaussian Mixture Models allow us to model robustly dynamic backgrounds. For foreground detection, the use of a linear saturation function avoids crisp decision in the classification. The fuzzy integrals (Sugeno and Choquet integrals) aggregate adequately the color and texture features to deal with illumination changes and shadows. Fuzzy adaptive learning rates permit us to adapt robustly the background to the changes that occur in videos. Future investigations may concern other issues of background subtraction:

- Fuzzy background modeling: T2-FGMMs allow us to deal with dynamic backgrounds but it is a parametric method, as are the crisp GMMs. It will be interesting to develop a fuzzy nonparametric method to model multimodal background. Furthermore, it would be interesting to use spatial information in the T2-FGMMs.

- Fuzzy foreground detection: Foreground detection is considered as a classification process with two classes, that is, background and foreground. Maybe it will be pertinent to add a third class for the shadows. Furthermore, feature selection using the fuzzy support vector machine can be applied to select the more discriminant features.

Acknowledgments

The author wants to acknowledge Lucia Maddalena (ICAR, National Research Council, Italy) and Alfredo Petrosino (DSA, University Parthenope of Naples, Italy) for providing the comparative results on the adaptive learning rate [55] and their algorithm [40, 41]. The author wants to thank Fida El Baf (Laboratoire MIA, University of La Rochelle, France) for providing her results on the Sugeno and Choquet integrals [3–5].

References

1. M. Azab, H. Shedeed, and A. Hussein. A new technique for background modeling and subtraction for motion detection in real-time videos. In *International Conference on Image Processing*, ICIP 2010, pages 3453–3456, September 2010.
2. F. El Baf and T. Bouwmans. Comparison of background subtraction methods for a multimedia learning space. In *International Conference on Signal Processing and Multimedia*, SIGMAP'07, July 2007.
3. F. El Baf, T. Bouwmans, and B. Vachon. Foreground detection using the Choquet integral. In *International Workshop on Image Analysis for Multimedia Interactive Integral*, WIAMIS 2008, pages 187–190, May 2008.
4. F. El Baf, T. Bouwmans, and B. Vachon. Fuzzy foreground detection for infrared videos. In *International Workshop on Object Tracking and Clas-*

sification in and beyond the Visible Spectrum, OTCBVS 2008, pages 1–6, June 2008.

5. F. El Baf, T. Bouwmans, and B. Vachon. Fuzzy integral for moving object detection. In *International Conference on Fuzzy Systems*, FUZZ-IEEE 2008, pages 1729–1736, June 2008.
6. F. El Baf, T. Bouwmans, and B. Vachon. Type-2 fuzzy mixture of Gaussians model: Application to background modeling. In *International Symposium on Visual Computing*, ISVC 2008, pages 772–781, December 2008.
7. F. El Baf, T. Bouwmans, and B. Vachon. Fuzzy statistical modeling of dynamic backgrounds for moving object detection in infrared videos. In *OTCBVS 2009*, pages 60–65, June 2009.
8. F. El Baf, T. Bouwmans, and B. Vachon. A fuzzy approach for background subtraction. In *International Conference on Image Processing*, ICIP 2008, pages 2648–2651, October 2008.
9. H. Bhaskar, L. Mihaylova, and A. Achim. Video foreground detection based on symmetric alpha-stable mixture models. *IEEE Transactions on Circuits, Systems and Video Technology*, March 2010.
10. H. Bhaskar, L. Mihaylova, and S. Maskell. Automatic target detection based on background modeling using adaptive cluster density estimation. *LNCS from the 3rd German Workshop on Sensor Data Fusion: Trends, Solutions, Applications*, September 2007.
11. B.Lee and M. Hedley. Background estimation for video surveillance. In *Image and Vision Computing New Zealand*, IVCNZ, pages 315–320, 2002.
12. T. Bouwmans. Subspace learning for background modeling: A survey. *RPCS*, 2(3):223–234, November 2009.
13. T. Bouwmans and F. El Baf. Modeling of dynamic backgrounds by type-2 fuzzy Gaussians mixture models. *MASAUM Journal of Basics and Applied Sciences*, 1(2):265–277, September 2009.
14. T. Bouwmans, F. El-Baf, and B. Vachon. Background modeling using mixture of Gaussians for foreground detection: A survey. *RPCS*, 1(3):219–237, November 2008.
15. D. Butler, V. Bove, and S. Shridharan. Real time adaptive foreground/background segmentation. *EURASIP*, pages 2292–2304, 2005.
16. J. Carranza, C. Theobalt, M. Magnor, and H. Seidel. Free-viewpoint video of human actors. *ACM Transactions on Graphics*, 22(3):569–577, 2003.
17. T. Chang, T. Ghandi, and M. Trivedi. Vision modules for a multi sensory bridge monitoring approach. In *ITSC 2004*, pages 971–976, 2004.
18. S. Cheung and C. Kamath. Robust background subtraction with foreground validation for urban traffic video. *Journal of Applied Signal Processing, EURASIP*, 2005.
19. D. Culbrik, O. Marques, D. Socek, H. Kalva, and B. Furht. Neural network approach to background modeling for video object segmentation. *IEEE Transaction on Neural Networks*, 18(6):1614–1627, 2007.
20. J. Davis and M. Goadrich. The relationship between precision-recall and roc curves. In *International Conference on Machine Learning*, ICML 2006,

pages 233–240, 2006.
21. M. Detyniecki. Fundamentals on aggregation operators. In *AGOP*, 2001.
22. Y. Ding, W. Li, T. Fan, and H. Yang. Robust moving object detection under complex background. *Computer Science and Information Systems*, 7(1), February 2010.
23. A. Elgammal and L. Davis. Non-parametric model for background subtraction. In *6th European Conference on Computer Vision, ECCV 2000*, pages 751–767, June 2000.
24. S. Elhabian, K. El-Sayed, and S. Ahmed. Moving object detection in spatial domain using background removal techniques — State-of-art. *RPCS*, 1(1):32–54, January 2008.
25. X. Fang, W. Xiong, B. Hu, and L. Wang. A moving object detection algorithm based on color information. In *International Symposium on Instrumentation Science and Technology*, 48:384–387, 2006.
26. J. Ferryman. http://www.cvg.rdg.ac.uk/pets2000/data.html. *PETS 2000*, 2000.
27. J. Ferryman. http://www.cvg.rdg.ac.uk/pets2006/data.html. *PETS 2006*, 2006.
28. M. Greiffenhagen, V. Ramesh, and H. Niemann. The systematic design and analysis cycle of a vision system: A case study in video surveillance. In *CVPR 2001*, 1(2):704, 2001.
29. M. Harville, G. Gordon, and J. Woodfill. Adaptive background subtraction using color and depth. In *IEEE International Conference on Image Processing*, ICIP 2001, October 2001.
30. M. Heikkila and M. Pietikinen. A texture-based method for modeling the background and detecting moving objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(4):657–662, 2006.
31. O. Javed, K. Shafique, and M. Shah. A hierarchical approach to robust background subtraction using color and gradient information. In *IEEE Workshop on Motion and Video Computing*, WMVC 2002, December 2002.
32. S. Kanpracha and S. Tangkawanit. Performance of RGB and HSV color systems in object detection applications under different illumination intensities. In *International Multi Conference of Engineers and Computer Scientists*, 2:1943–1948, March 2007.
33. H. Kashani, S. Seyedin, and H. Yazdi. A novel approach in video scene background estimation. *International Journal of Computer Theory and Engineering*, 2(2):274–282, April 2010.
34. K. Kim, T. Chalidabhongse, D. Harwood, and L. Davis. Real time foreground background segmentation using codebook model. *Real Time Imaging*, 11(3):167–256, 2005.
35. F. Kristensen, P. Nilsson, and V. Wall. Background segmentation beyond RGB. In *ACCV 2006*, pages 602–612, 2006.
36. L. Li and W. Huang. http://perception.i2r.astar.edu.sg/bkmodel/bkindex.html.
37. L. Li and W. Huang. Statistical modeling of complex background for foreground object detection. *IEEE Transaction on Image Processing*,

13(11):1459–1472, November 2004.

38. L. Maddalena and A. Petrosino. A self organizing approach to background subtraction for visual surveillance applications. *IEEE Transactions on Image Processing*, 17(7):1729–1736, 2008.
39. L. Maddalena and A. Petrosino. Multivalued background/foreground separation for moving object detection. In *International Workshop on Fuzzy Logic and Applications*, WILF 2009, 5571:263–270, June 2009.
40. L. Maddalena and A. Petrosino. Self organizing and fuzzy modelling for parked vehicles detection. In *Advanced Concepts for Intelligent Vision Systems*, ACIVS 2009, LNCS 5807, pages 422–433, 2009.
41. L. Maddalena and A. Petrosino. A fuzzy spatial coherence-based approach to background/foreground separation for moving object detection. *Neural Computing and Applications*, Springer London, 19:179–186, 2010.
42. N McFarlane, C. Schofield. Segmentation and tracking of piglets in images. *British Machine Vision and Applications*, pages 187–193, 1995.
43. S. Messelodi, C. Modena, N. Segata, and M. Zanin. A Kalman filter based background updating algorithm robust to sharp illumination changes. In *ICIAP 2005*, 3617:163–170, September 2005.
44. Y. Narukawa and T. Murofushi. Decision modelling using the Choquet integral. *Modeling Decisions for Artificial Intelligence*, 3131:183–193, 2004.
45. Y. Ohta, T. Kanade, and T. Sakai. Color information for region segmentation. *Computer Graphics and Image Processing*, 13(3):222–241, 1980.
46. D. Pokrajac and L. Latecki. Spatiotemporal blocks-based moving objects identification and tracking. *IEEE Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pages 70–77, October 2003.
47. H. Ribeiro and A. Gonzaga. Hand image segmentation in video sequence by GMM: A comparative analysis. In *XIX Brazilian Symposium on Computer Graphics and Image Processing*, SIBGRAPI 2006, pages 357–364, 2006.
48. J. Rossel-Ortega, G. Andrieu, F. Lopez-Garcia, and V. Atienza-Vanacloig. Background modeling with motion criterion and multi-modal support. In *International Conference on Computer Vision Theory and Application*, VISAPP 2010, May 2010.
49. J. Rossel-Ortega, G. Andrieu, A. Rodas-Jorda, and V. Atienza-Vanacloig. Background modeling in demanding situations with confidence measure. In *International Conference on Computer Vision*, ICPR 2008, December 2008.
50. J. Rossel-Ortega, G. Andrieu, A. Rodas-Jorda, and V. Atienza-Vanacloig. A combined self-configuring method for object tracking in colour video. In *International Conference on Computer Vision*, ICPR 2010, August 2010.
51. M. Shakeri and H. Deldari. Fuzzy-cellular background subtraction technique for urban traffic applications. *World Applied Sciences Journal*, 5(1), 2008.
52. M. Shakeri, H. Deldari, H. Foroughi, A. Saberi, and A. Naseri. A novel fuzzy

background subtraction method based on cellular automata for urban traffic applications. In *9th International Conference on Signal Processing*, ICSP 2008, pages 899–902, October 2008.

53. Y. Sheikh. http://www.cs.ucf.edu/yaser/backgroundsub.htm.
54. M. Sigari. Fuzzy background modeling/subtraction and its application in vehicle detection. In *World Congress on Engineering and Computer Science*, WCECS 2008, October 2008.
55. M. Sigari, N. Mozayani, and H. Pourreza. Fuzzy running average and fuzzy background subtraction: Concepts and application. *International Journal of Computer Science and Network Security*, 8(2):138–143, 2008.
56. M. Sivabalakrishnan and D. Manjula. Adaptive background subtraction in dynamic environments using fuzzy logic. *International Journal of Image Processing*, 4(1), 2010.
57. C. Stauffer and E. Grimson. Adaptive background mixture models for real-time tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 246–252, 1999.
58. M. Sugeno and S. Kwon. A new approach to time series modeling with fuzzy measures and the Choquet integral. In *IEEE International Conference on Fuzzy Systems*, pages 799–804, March 1995.
59. H. Tahani and J. Keller. Information fusion in computer vision using the fuzzy integral. In *IEEE Transaction on Systems, Man, and Cybernetics*, 20(3):733–741, 1990.
60. K. Toyama, J. Krumm, B. Brumiit, and B. Meyers. Wallflower: Principles and practice of background maintenance. In *International Conference on Computer Vision*, pages 255–261, September 1999.
61. C. Wren and A. Azarbayejani. Pfinder: Real-Time Tracking of the Human Body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):780–785, July 1997.
62. J. Zeng, L. Xie, and Z. Liu. Type-2 fuzzy Gaussian mixture. *Pattern Recognition*, 41(2):3636–3643, September 2009.
63. H. Zhang and D. Xu. Fusing color and texture features for background model. In *Third International Conference on Fuzzy Systems and Knowledge Discovery*, FSKD'06, 4223(7):887–893, September 2006.
64. J. Zheng, Y. Wang, N. Nihan, and E. Hallenbeck. Extracting roadway background image: A mode based approach. *Journal of Transportation Research Report*, 1944:82–88, 2006.