Artificial Intelligence Evolution
http://ojs.wiserpub.com/index.php/AIE/

UNIVERSAL WISER PUBLISHER

Research Article

Weak-Strong Self-Adapting Fuzzy Neural Classifier for Dynamic Object Detection in RGBD Videos

Mario I. Chacon-Murguia[1*], Huber E. Orozco-Rodríguez[2], Graciela Ramirez-Alonso[3], Juan A. Ramirez-Quintana[1]

[1]Chihuahua Institute of Technology, Chihuahua, Mexico
[2]Visual and Parallel Computing Group, Intel Tecnologia de Mexico, Guadalajara, Mexico
[3]Faculty of Engineering, Autonomous University of Chihuahua, Chihuahua, Mexico
Email: mario.cm@chihuahua.tecnm.mx

Received: 15 October 2022; **Revised:** 15 December 2022; **Accepted:** 16 December 2022

Abstract: This paper presents a fuzzy neural method to model background from videos in order to detect dynamic objects. The method includes a weak fuzzy classifier that performs an initial foreground and background separation based on color and depth differences between the actual frame and background models. The outputs of this fuzzy system are weighted according to the result of the color and depth noise modeling. A degree of uncertainty and the strength of decisions, in combination with the weighting results, are used by the method to define more accurately the dynamic objects through a strong fuzzy classifier. The final stage of foreground detection is implemented with a Discrete-Time Cellular Neural Network to improve the foreground definition. Finally, the color and depth background models are updated based on a fuzzy learning rate strategy. The method was evaluated with the new SBM-RGBD database and compared against several state-of-the-art methods showing a similar or better performance considering the quantitative and qualitative evaluations.

***Keywords*:** RBGD videos, background model, dynamic objects, fuzzy neural classifier

1. Introduction

The identification of dynamic objects in video sequences has gained much attention from researchers because it is the base of different and sophisticated applications such as autonomous driving, medical care, rehabilitation, and surveillance systems, among others [1-5]. With the introduction of low-cost depth cameras, some algorithms have included in their models the analysis of depth information [6-7]. By the use of depth maps, the false positive detections in the foreground caused by only considering color information, such as color camouflage, dynamic background, and illumination changes, could be reduced [8]. In a depth map, the pixel information is proportional to the distance from the device to the objects in the scene. Even when color camouflage could be reduced by considering depth information, there is another issue known as depth camouflage. Depth camouflage is produced when the background and foreground are close in depth. Also, there are some other problems caused by the use of depth sensors, such as the lack of depth information in some pixels defined as no-measured pixels, the irregular definition of object boundaries, the low sensitivity to long distances, and the fact that objects near to the sensor may not contain information [9].

DOI: https://doi.org/10.37256/aie.4120232049

Some applications where depth sensors are used to separate foreground and background regions are presented below. Kwolek and Kepski [10] implemented a human fall detection system based on accelerometer data and depth maps. If the measured acceleration surpasses a threshold value, the system extracts the person, calculates features, and a classifier activates an alarm if this is the case. Xue et al. [2] developed a human tracking system where the crowd is known in advance, or all persons have appeared from the beginning. A motion model based on spatial and kinetic features in combination with a deep convolutional neural network tracks people in the scene. Camplani and Salgado [11] combined the results of two weak Bayesian classifiers to identify dynamic objects in video sequences. One classifier is based on depth features, and the other is on color information. The final foreground detection is obtained through a weighted average of the two classifiers. Their method was validated with the RGB-D object detection dataset. This dataset comprises five indoor sequences, mainly considering cast shadows and color and depth camouflage. In another work [8], a non-parametric kernel density estimation (KDE) approach was proposed based on depth and color information. In order to validate their method, the authors developed the GSM dataset. This dataset has seven sequences and considers color and depth camouflage, illumination changes, shadows, walking objects, and bootstrapping issues. Trabelsi et al. [9] proposed a KDE model in combination with a Gauss transform. In order to prove the robustness of the proposed model, it was evaluated with four databases achieving accurate segmentation results. Sultana et al. [12] implemented a generative adversarial network with RGB-D data conditioned on ground-truth information to segment the foreground. The network was trained to distinguish between real vs. fake foreground samples, and during testing, the network generated the foreground results considering two different datasets. In [13], two foreground segmentation algorithms are presented: a Gaussian Mixture Model (GMM) and a Pixel-Based Adaptive Segmenter (PBAS). These algorithms were adapted to work with two RGB-D sensors and a publicly available dataset. An increase in segmentation accuracy was observed when using RGB-D data.

As can be observed, one of the issues related to the validation of different segmentation methods is that most of the time, different authors use different data sets. This process makes it difficult to perform a fair comparison between them. However, an important effort to surpass this issue is to use the new dataset called SBM-RGBD. SBM-RGBD is a dataset introduced in 2017 [14-15] organized by Massimo Camplani, University of Bristol, UK, Lucia Maddalena, National Research Council, Italy, Gabriel Moyà Alcover, Universitat de les Illes Balears, Spain, Alfredo Petrosino, University of Naples Parthenope, Italy, and Luis Salgado, Universidad Politécnica de Madrid & Universidad Autónoma de Madrid, Spain. This dataset considers videos from a collection of different public datasets, the GSM dataset [8], MULTIVISION [16], Princeton Tracking Benchmark [17], RGB-D object detection dataset [11], and UR Fall Detection Dataset [10]. Therefore, SBM-RGBD constitutes an excellent medium to evaluate algorithms to detect dynamic objects using color and depth information. Furthermore, this dataset has been used in [18-21] to validate their background models.

Considering the previous issues, the present paper proposes a fuzzy neural classifier that considers color and depth information. Our method is named FN-DTCNM, a fuzzy neural discrete-time cellular neural network model. In this model, a weak fuzzy classifier performs an initial foreground and background separation based on color and depth differences between the actual frame and background models. The outputs of this fuzzy system are weighted according to the result of the color and depth noise modeling. A degree of uncertainty and the strength of decisions, in combination with the weighting results, define more accurately dynamic objects. This analysis is performed with a strong fuzzy classifier. The final stage of foreground detection is implemented with a Discrete-Time Cellular Neural Network (DTCNN) to improve the foreground definition. This network considers the membership grades of color and depth differences of the pixel under analysis and its neighbor to eliminate false positive detections. Once the pixels are classified, the color and depth background models are updated based on a fuzzy learning rate strategy. Our method was validated with the SBM-RGBD dataset achieving very competitive results considering quantitative and qualitative evaluations.

The rest of the paper is organized as follows. Section 2 presents the fuzzy neural background subtraction method, and the evaluation and results with depth databases are reported in Section 3. Finally, Section 4 presents the conclusions.

2. Fuzzy neural background subtraction method

Figure 1 presents a block diagram of the FN-DTCNM model. The input is an RGBD frame, where $I_C(x, y, t)$ is the

color component, $I_D(x, y, t)$ is the depth component, x, y is the pixel position, and t is the time index. The initial color and depth frame of the video sequence is used to generate the first color and depth background models. The modules of the weak fuzzy classifier that separates the initial foreground and background regions are indicated in blue. The outputs of this system are analyzed and pondered by a weighted strategy that considers color and depth noise. These sections are marked in green. The second fuzzy classifier (represented in brown) separates in a more precise way the identification of the foreground. The DTCNN module, in yellow, performs a reduction of false positive detections by considering the fuzzy membership grades of the previous analysis. The final stage represents the fuzzy learning rate strategy implemented to update the color and depth background models. This part is marked in purple. A detailed description of each module is presented in the following sections.

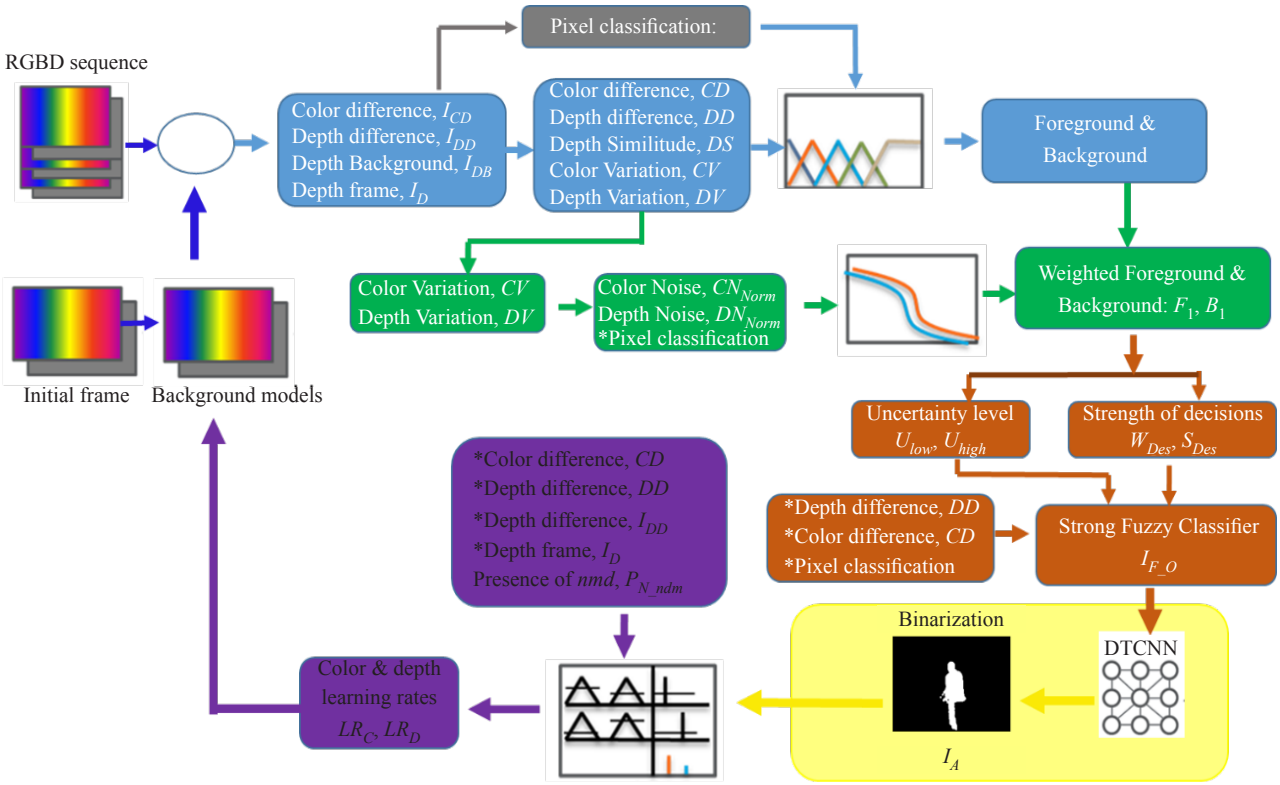

Figure 1. Block diagram of the FN-DTCNM model

* The asterisk in the figure means that this particular input was defined in a previous module.

An important aspect of the proposed system is its self-adapting capability. This self-adapting capability is achieved through several auto-adapting parameters that are briefly described next and specifically explained in specific sections. The parameter σ_D, the standard deviation of the depth measurements, defined in Section 2.1 Background model initialization, helps to adjust the membership functions that represent the Depth difference and Depth variation. In Section 2.2 Weak fuzzy classifier, the Entropy of the actual color frame E_C, and the Entropy of the color background E_{CB}, are compared, and their difference is defined as Entropy change Δ_E. This value is used to calculate the automatic displacement, Δ_C, of the fuzzy variable *color difference*. Δ_C is a parameter of membership functions, making the system more robust to illumination changes. That is, it provides the self-adapting capability to deal with illumination changes. Another parameter that provides self-adapting capabilities to the systems is the parameter σ_F defined in Section 2.2 Weak fuzzy classifier, specifically in the level of Depth similitude.

Weighting parameters that also provide self-adapting to the systems are explained in section 2.3, Weighting of the fuzzy classifier. For example, in order to increase the reliability of the fuzzy classifier, it is included a weighting

function on its output. A temporal analysis of color and depth variations is used to calculate this weighting criterion. The color noise modeling $CN(x, y, t)$ for each pixel is calculated by a Hebbian rule. Also, the level of depth noise $DN(x, y, t)$ is automatically updated by a Hebbian rule. At the same time, $CN(x, y, t)$ and $DN(x, y, t)$ are employed to update the weights of the fuzzy classifier, which is also a self-adapting mechanism.

The sections indicated in the previous text provide a specific explanation of each parameter that provides self-adapting capabilities.

2.1 *Background model initialization*

The initial frame of the RGBD sequence is considered to initialize the color and depth background models. Let $I_C(x, y, 0)$ and $I_D(x, y, 0)$ define the first color and depth frames of the video sequence. $I_{CB}(x, y, 1)$ and $I_{DB}(x, y, 1)$ will describe the initial color and depth background models.

The color space used in this work is the Hue Saturation Value (HSV), and the depth map corresponds to the distance information in millimeters. The Euclidean distance between the current frame and background models represents color and depth differences and are defined by I_{CD} and I_{DD}, respectively.

In order to reduce possible errors in the values of I_{DD}, it is obtained without considering no-measured depth (*nmd*) pixels. *nmd* pixels will be managed by the fuzzy algorithm.

A way to reduce the false positive detection in the foreground segmentation is by modeling the noise of the depth map. Nguyen et al. exposed in [22] the existence of oscillations in the depth measurements obtained with the Kinect sensor. The standard deviation of these oscillations increased by a quadratic factor in relation to the distance between the sensor and the measured objects. These oscillations may cause false positive detections in the segmentation results. Nguyen modeled the standard deviation of the depth measurements as follows

$$\sigma_D\left(x,y,t\right) = 0.0012 + 0.0019\left(I_{DB}\left(x,y,t\right) - 0.4\right)^2 \tag{1}$$

With this noise modeling, it is possible to obtain an interval $[-\sigma_D(x, y, t), +\sigma_D(x, y, t)]$ where the depth information has low levels of noise.

2.2 *Weak fuzzy classifier*

Similar to the work presented in [23], the illumination changes in the video sequence were detected by an Entropy, E, analysis. The Entropy of the actual color frame E_C, and the Entropy of the color background E_{CB}, are compared, and their difference is defined as Entropy change, Δ_E. This value is used to calculate the automatic displacement of the fuzzy variable *color difference* $\Delta_C = \min(0.9|\Delta_E|, 0.9)$.

The inputs to the weak fuzzy system are extracted from the color and depth differences (calculated by Eq. (2)), depth background model, and depth input frame. The fuzzy variables defined are the *color difference* (CD), *depth difference* (DD), *depth similitude* (DS), *color variation* (CV), and *depth variation* (DV). Their linguistic representations are described below.

Color difference: The color difference variable has three fuzzy values: *small*, *medium*, and *big* $\{CD_S, CD_M, CD_B\}$. CD_S is defined as a sigmoidal function $Sigmoidal(I_{CD}(x, y, t), -30, 0.1 + \Delta_C)$, CD_M uses a Gaussian function $Gaussian(I_{CD}(x, y, t), 0.05, 0.15 + \Delta_C)$ and CD_B is a sigmoidal function defined as $Sigmoidal(I_{CD}(x, y, t), 30, 0.2 + \Delta_C)$. As previously explained, the parameter Δ_C is included to increase the robustness of the system to illumination changes. The fuzzy value CD_S mainly considers pixels with color differences barely perceived. CD_M deals mainly with noisy measurements and shadows. CD_B includes highly perceptive color differences. When a drastic illumination change is detected in the scene, the increment Δ_C for the color difference value causes a maximum displacement of 0.9 on its membership functions. With this increment in the CD_S sigmoidal function, all the color difference values will be evaluated with this function. Figure 2 shows the membership functions of the color difference variable with a $\Delta_C = 0$.

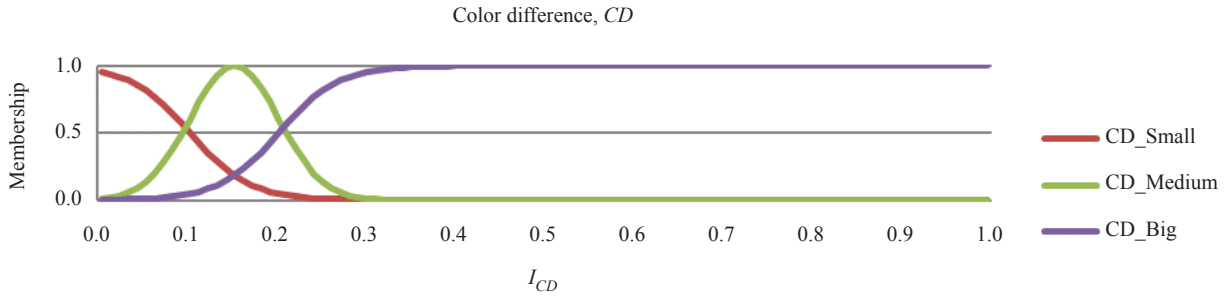

Figure 2. Membership functions of the color difference variable where $\Delta_C = 0$

Depth difference: To represent the depth difference fuzzy values, the values were defined as *small*, *medium*, and *big* $\{DD_S, DD_M, DD_B\}$. DD_S is defined with a sigmoidal function $Sigmoidal(I_{DD}(x, y, t)/\sigma_D(x, y, t), -5, -2)$, DD_M uses a sigmoidal function $Dsigmoidal(I_{DD}(x, y, t)/\sigma_D(x, y, t), 5, -4, 5, -1)$, and DD_B is defined as $Sigmoidal(I_{DD}(x, y, t)/\sigma_D(x, y, t), -5, -3)$. As can be observed, the depth difference value is normalized with respect to the standard deviation $\sigma_D(x, y, t)$. Therefore, the universe of discourse of this fuzzy variable is in terms of standard deviations. Figure 3 shows the membership functions of this variable.

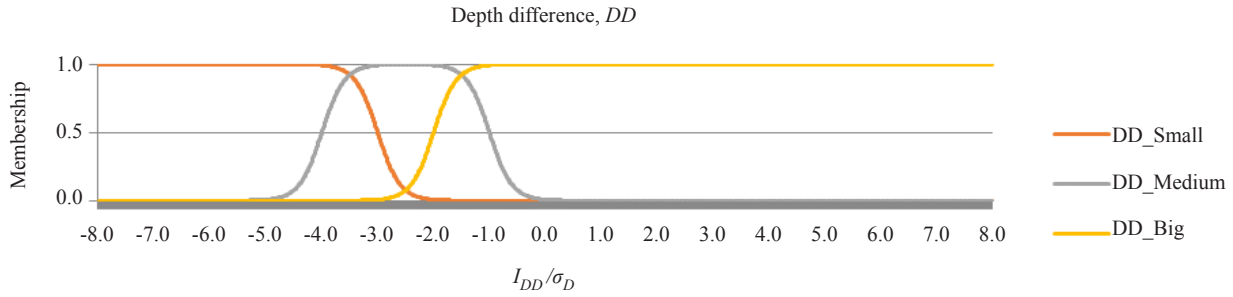

Figure 3. Membership functions of the depth difference variable

Depth similitude: The third fuzzy variable, depth similitude (DS), describes the similarity between the current depth frame and the depth background model. In order to calculate this similitude variable, it is necessary to filter the depth background model I_{DB} by an average filter of order 9, yielding I_{FDB}. This filtered depth background model has variations with respect to the original model. Therefore, it is necessary to calculate a new standard deviation to include the pixels generated by the filter and define new intervals $[-\sigma_F(x, y, t), \sigma_F(x, y, t)]$

$$\sigma_F(x,y,t) = 0.0012 + 0.0019\left(I_{FDB}(x,y,t) - 0.4\right)^2 \tag{2}$$

The level of similitude $I_S(x, y, t)$ is the input to the *depth similitude* fuzzy value described as

$$I_S(x,y,t) = \frac{\left|I_D(x,y,t) - I_{FDB}(x,y,t)\right|}{\sigma_F(x,y,t)} \tag{3}$$

DS has two fuzzy values, *low* and *high* similitude $\{DS_L$ and $DS_H\}$. DS_L and DS_H are defined with sigmoidal functions DS_L $Sigmoidal(I_S(x, y, t), 10, 4)$ and DS_H $Sigmoidal(I_S(x, y, t), -10, 4)$. Figure 4 shows the definition of the *DS*

fuzzy value and its membership functions.

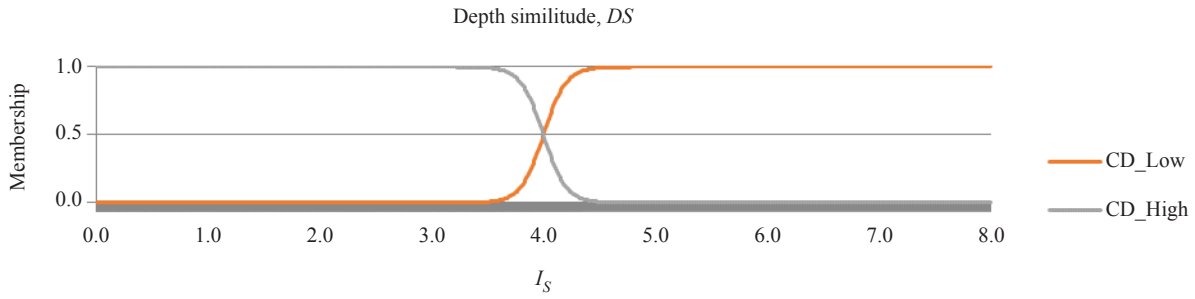

Figure 4. Membership function of the depth similitude variable

Color Variation: The fuzzy system also handles variations of color and depth information between the actual frames and background models. The fuzzy variable *color variation* (CV) has two fuzzy values: *small* and *big* $\{CV_S, CV_B\}$. The input to this variable is the color difference, I_{CD}. In this fuzzy variable, there were implemented two sigmoidal functions CV_S *Sigmoidal*($I_{CD}(x, y, t)$, –50, 0.1) and CV_B *Sigmoidal*($I_{CD}(x, y, t)$, 50, 0.1). Figure 5 shows the membership functions defined for this fuzzy variable. The variable *Color difference* evaluates the same input, I_{CD}, but with three fuzzy values: *small*, *medium*, and *big* $\{CD_S, CD_M, CD_B\}$. Because of the definition of the fuzzy values in CV and CD, all the inputs mapped to CV_S will be the same as CD_S, whereas the inputs mapped to CV_B will correspond to those mapped in CD_M and CD_B.

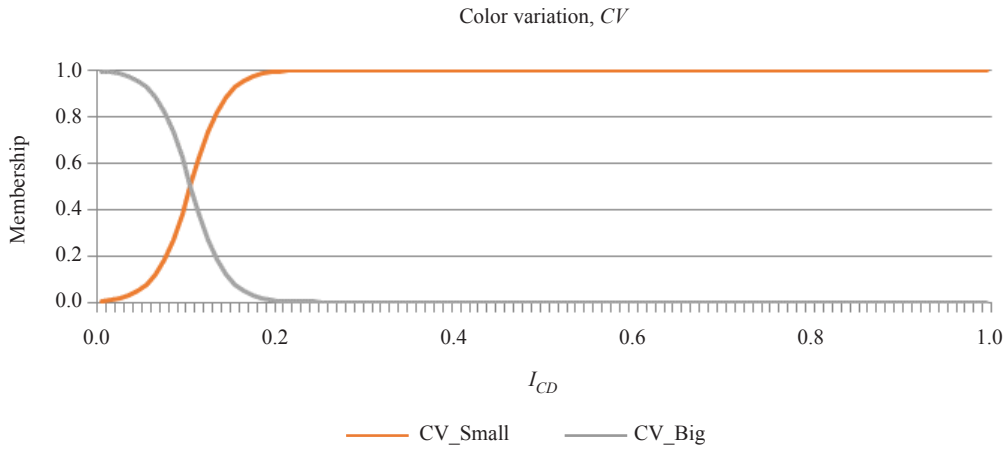

Figure 5. Membership function of the color variation variable

Depth variation: The fuzzy variable depth variation (DV) has two fuzzy values: *small* and *big* $\{DV_S, DV_B\}$ defined as follows: DV_S *Sigmoidal*($|I_{DD}(x, y, t)/\sigma_D(x, y, t)|$, –1, –4) and DV_B *Sigmoidal*($|I_{DD}(x, y, t)/\sigma_D(x, y, t)|$, 1, –4). The input to this fuzzy variable is the absolute value of the depth difference divided by the standard deviation of the depth model. Figure 6 presents the definition of these membership functions graphically.

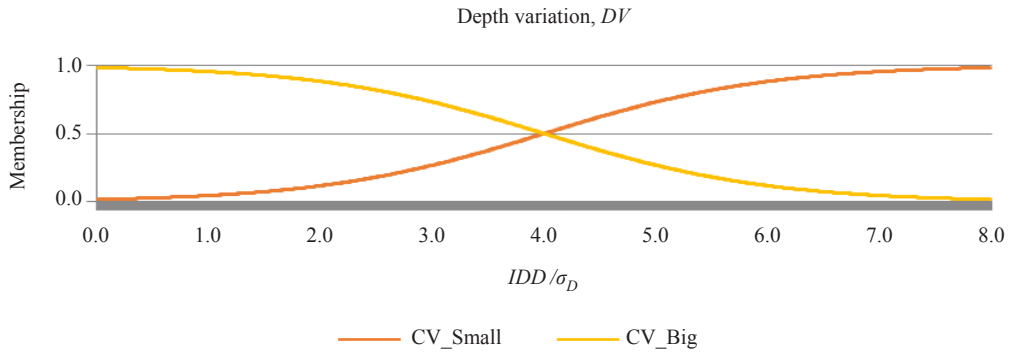

Figure 6. Membership function of the depth variation variable

Before the definition of the fuzzy rules of the system, it is necessary to classify each pixel according to its depth information. Four linguistic descriptors were used to classify them. A pixel is defined as a *complete pixel* if it has depth information in the background depth model and the current frame. An *empty pixel* is defined as a pixel with information in the background depth model but not in the current frame. A *new pixel* is a pixel with information in the current depth frame but not in the background model. A *color pixel* is a pixel that does not have depth information in the current frame or the background model.

The rules of this fuzzy system are presented below. The antecedent weights are determined to distinguish the importance of the foreground pixels when it is necessary and were defined experimentally.

$$\begin{aligned}
&\text{If } CD \text{ is } CD_S \ OR \ CD \text{ is } CD_M * 0.3 \text{ Then } i_C(x,y,t) \text{ is } Background\\
&\text{If } CD \text{ is } CD_M * 0.7 \ OR \ CD \text{ is } CD_B \text{ Then } i_C(x,y,t) \text{ is } Foreground\\
&\text{If } DD \text{ is } DD_S \text{ OR } DD \text{ is } DD_M * 0.6 \text{ Then } i_D(x,y,t) \text{ is } Background\\
&\text{If } DD \text{ is } DD_M * 0.4 \text{ OR } DD \text{ is } DD_B \text{ Then } i_D(x,y,t) \text{ is } Foreground\\
&\text{If } DS \text{ is } DS_H \text{ Then } i_D(x,y,t) \text{ is } Background\\
&\text{If } i_D(x,y,t) \text{ is } EmptyPixel \text{ Then } i_D(x,y,t) \text{ is } Background\\
&\text{If } i_D(x,y,t) \text{ is } NewPixel \text{ Then } i_D(x,y,t) \ NO \text{ is } Background\\
&\text{If } i_D(x,y,t) \text{ is } NewPixel \text{ Then } i_D(x,y,t) \text{ is } Foreground
\end{aligned} \tag{4}$$

where i_C stands for the color pixel membership value of the *Background* or *Foreground* classes. Similarly, i_D stands for the depth pixel membership value for the *Background* or *Foreground* classes.

2.3 *Weighting of the fuzzy classifier*

In order to increase the reliability of the fuzzy classifier, a weighting function was included in its output. A temporal analysis of color variations and depth variations is used to calculate this weighting criterion. First, the membership value of the *big color variation* (μ_{CVB}) variable is analyzed as follows

$$\begin{aligned}
\Delta_{CV_B}(x,y,t-1) &= \left|\mu_{CV_B}(x,y,t-1) - \mu_{CV_B}(x,y,t-2)\right|\\
\Delta_{CV_B}(x,y,t) &= \left|\mu_{CV_B}(x,y,t) - \mu_{CV_B}(x,y,t-1)\right|
\end{aligned} \tag{5}$$

Where Δ_{CVB} stands for an increment in color variation. The color noise modeling $CN(x, y, t)$ for each pixel is calculated by the Hebbian rule as

$$CN(x,y,t) = CN(x,y,t-1)(1-\alpha_{CN}) + \Delta_{CV_B}(x,y,t)\Delta_{CV_B}(x,y,t-1)\beta_{CN} \quad (6)$$

where the variables α_{CN} and β_{CN} describe the decay and learning rate parameters. Similarly, the absolute difference of the *big depth variation* variable (μ_{DVB}) is computed at times t and $t-1$, Δ_{DVB}, and times $t-1$ and $t-2$, Δ_{DVB}. The level of the depth noise $DN(x, y, t)$ is also computed for $CN(x, y, t)$ but using the corresponding terms of DN. The color and depth noise modeling results are normalized in relation to their decay and learning rate parameters resulting in variables whose ranges vary between [0, 1] and are denoted by DN_{Norm} and DN_{Norm}. In our model, these values are considered membership grades. The values defined for the different decay and learning rate parameters are $\alpha_{CN} = \alpha_{DN} = 0.1$, $\beta_{CN} = \beta_{DN} = 1.0$. These values were obtained experimentally.

The weighting of the fuzzy classifier is defined by two sigmoidal functions $Sigmoidal(CN_{Norm}(x, y, t), -30, 0.15)$ and $Sigmoidal(DN_{Norm}(x, y, t), -20, 0.3)$ respectively, illustrated in Figure 7.

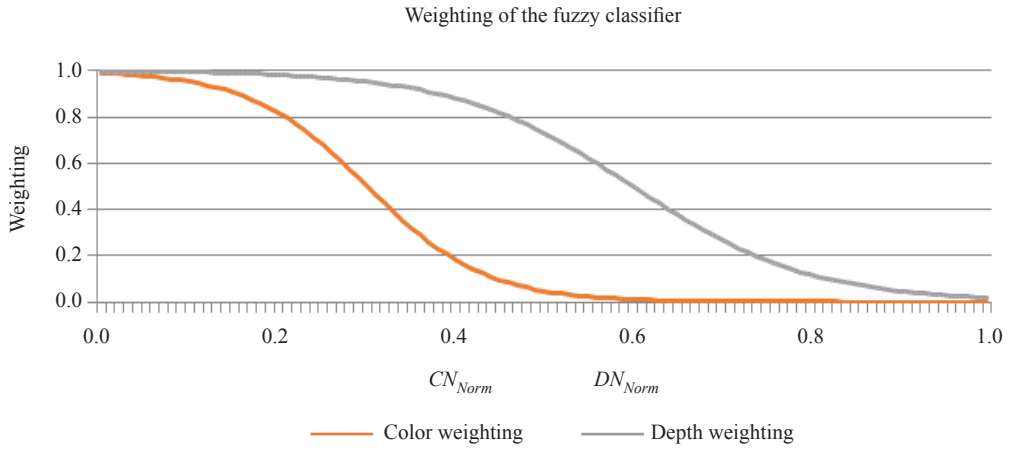

Figure 7. Membership functions of the weighting parameter

The weighting color function, W_C, considers as input the normalized color noise CN_{Norm} whereas the input of the weighting depth function, W_D, is the normalized depth noise DN_{Norm}. The parameters of these membership functions were defined experimentally. In order to cope with illumination changes, the center of the weighting color function, $W_C(x, y, t)$, is displaced by a factor of $(1 - \Delta_C)$.

The pixel classification into foreground (F_1) or background (B_1) by considering the weighting criterion is performed as follows

$$\begin{aligned} B_1(x,y,t) &= W_C(x,y,t)i_c(x,y,t) \quad \text{if } i_C(x,y,t) \text{ is } Background \\ F_1(x,y,t) &= W_C(x,y,t)i_c(x,y,t) \quad \text{if } i_C(x,y,t) \text{ is } Foreground \end{aligned} \quad \forall(EmptyPixel \vee ColorPixel) \quad (7)$$

$$B_1(x,y,t) = \frac{W_C(x,y,t)i_c(x,y,t) + W_D(x,y,t)i_d(x,y,t)}{W_C(x,y,t) + W_D(x,y,t)} \text{ if } i_C(x,y,t) \text{ and } i_d(x,y,t) \text{ are } Background$$

$$F_1(x,y,t)=\frac{W_C(x,y,t)i_c(x,y,t)+W_D(x,y,t)i_d(x,y,t)}{W_C(x,y,t)+W_D(x,y,t)} \text{ if } i_C(x,y,t) \text{ and } i_d(x,y,t) \text{ are } Foreground$$

$$\forall\ (CompletePixel \vee NewPixel)$$

2.4 *Strong fuzzy classifier*

The segmentation results of Eq. (7) are prone to errors caused by color and depth camouflage. For this reason, there was included an analysis of the uncertainty level and the strength of the decisions between the outputs of Eq. (7). For example, there could exist errors in the pixel classification when the difference between F_1 and B_1 is low. Therefore, an *uncertainty level* detection (U) is computed represented by the difference between F_1 and B_1 for μ_{Ulow}, and its complement is defined as μ_{Uhigh}.

When there exist high levels of color and depth noise in the scene, the results in (7) are very low. In this case, it is necessary to include an analysis of the strength of the decisions taken by the classifier and define them as weak decisions, W_{Des}, or strong decisions, S_{Des} represented by *Sigmoidal*(max($F_1(x, y, t)$, $B_1(x, y, t)$), –20, 0.25) and *Sigmoidal*(max($F_1(x, y, t)$, $B_1(x, y, t)$), 20, 0.25), respectively.

Figure 8 shows the graphical representation of this analysis. When the maximum output of Eq. (7) is less than 0.25, it will be considered a weak decision. On the other hand, if the output surpasses this value, it is defined as a strong decision.

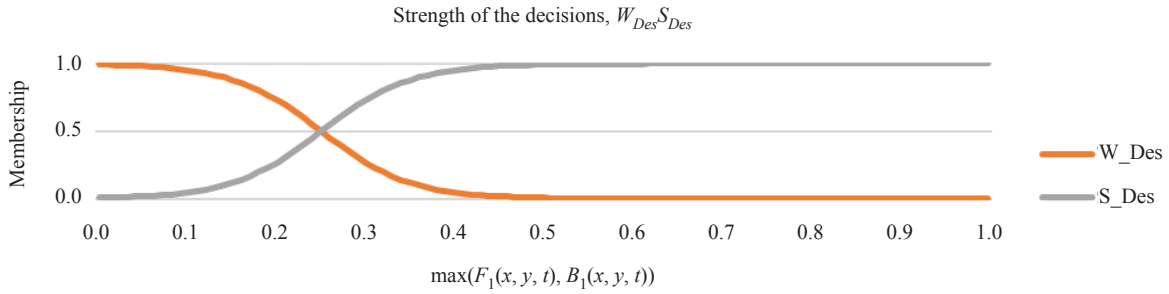

Figure 8. Membership functions of the strength analysis

Based on the previous issues, three rules were included to consider the previous uncertainty level analysis and the strength of the decisions.

Rule 1: When a weak decision is detected in a pixel classification, it is a good option to classify it as background. This expression is represented by

$$\text{If } i(x,y,t) \text{ is } W_{Des} \text{ Then } i(x,y,t) \text{ is Background Else } i(x,y,t) \text{ stay the same} \tag{8}$$

In terms of membership functions, this previous relation is expressed as

$$B_1(x,y,t)=\mu_{WDes}(x,y,t)+\mu_{SDes}(x,y,t)B_1(x,y,t)$$

$$F_1(x,y,t)=\mu_{SDes}(x,y,t)F_1(x,y,t) \tag{9}$$

Rule 2: The next fuzzy rule applies to pixels classified as complete with *small* depth difference (DD_S) and *high* uncertainty level (U_{high})

If U is $U_{high} \wedge DD$ is DD_S Then $i(x,y,t)$ is Background Else $i(x,y,t)$ stay the same; $\forall$ *completePixel* (10)

The following equations represent the above function in terms of membership functions,

$$B_1(x,y,t) = \mu_{Uhigh}(x,y,t)\,\mu_{DD_S}(x,y,t) + \left(1 - \mu_{Uhigh}(x,y,t)\,\mu_{DD_S}(x,y,t)\right) B_1(x,y,t)$$

$$F_1(x,y,t) = \left(1 - \mu_{Uhigh}(x,y,t)\,\mu_{DD_S}(x,y,t)\right) F_1(x,y,t)$$

$$\forall\ CompletePixel \tag{11}$$

Rule 3: The last fuzzy rule is applied to empty or color pixels with *high* uncertainty levels (U_{high}) and *small* color differences (CS_s)

If U is $U_{high} \wedge CD$ is VERY CD_S Then $i(x,y,t)$ is Background Else $i(x,y,t)$ stay the same

$$\forall\ \left(EmptyPixel \vee ColorPixel\right) \tag{12}$$

In terms of membership functions, this expression is represented as

$$B_1(x,y,t) = \mu_{Uhigh}(x,y,t)\,\mu_{CD_S}(x,y,t)^2 + \left(1 - \mu_{Uhigh}(x,y,t)\,\mu_{CD_S}(x,y,t)^2\right) B_1(x,y,t)$$

$$F_1(x,y,t) = \left(1 - \mu_{Uhigh}(x,y,t)\,\mu_{CD_S}(x,y,t)^2\right) F_1(x,y,t)$$

$$\forall\ \left(EmptyPixel \vee ColorPixel\right) \tag{13}$$

The final binary image that represents the dynamic objects is termed I_{F_O}.

2.5 *Discrete cellular neural network*

Once the dynamic object detection has been achieved, some algorithms perform a post-processing analysis using morphologic operators in order to reduce false positives. This morphological process eliminates some of these errors. However, it also eliminates part of the dynamic objects and will not reduce large blobs detected as false positives caused by incorrect depth measurements. A more accurate solution will be obtained by implementing a discrete cellular neural network (DTCNN) that considers the neighborhood membership grades of a pixel in the binarization process [23]. The DTCNN implemented is described as

$$\chi(x,y,n) = \sum_{h=-r}^{r}\sum_{j=-r}^{r} A(h,j)\,y_C(x+h,y+j,n) + \sum_{h=-r}^{r}\sum_{j=-r}^{r} B(h,j)\,I_{F\_0}(x+h,y+j) + Z \tag{14}$$

where $\chi(x, y, n)$ is the DTCNN state at iteration n. The kernel coefficients $A(h, j)$ define the state of the neuron (x, y) based on the output $y_C(x, y, n)$, and the kernel coefficients $B(h, j)$ describe how the state of the neuron depends on the input $I_{F\_O}(x, y)$, and Z is the neuron polarization used to adjust its threshold. The output of the implemented network

$y_C(x, y, n)$ is defined by the state $\chi(x, y, n)$

$$y_C(x,y,n+1) = 0.5\left(\left|\chi(x,y,n)+1\right| - \left|\chi(x,y,n)-1\right|\right) \tag{15}$$

The initial state of the network $\chi(x, y, 0)$ depends on the membership functions of the fuzzy classification of pixels as described below

$$\chi(x,y,0) = \begin{cases} 1-2F_1(x,y,t) & \text{if } I_{F\_O} = 1 \\ 1-2B_1(x,y,t) & \text{if } I_{F\_O} = 0 \end{cases} \tag{16}$$

After n iterations of the network, its output is binarized as described in Eq (17), where the sign of the output defines the final state of the pixel

$$I_O(x,y,t) = \begin{cases} 1 & \text{if } \left(1-y_C(x,y,n)\right)/2 > 0.5 \\ 0 & \text{otherwise} \end{cases} \tag{17}$$

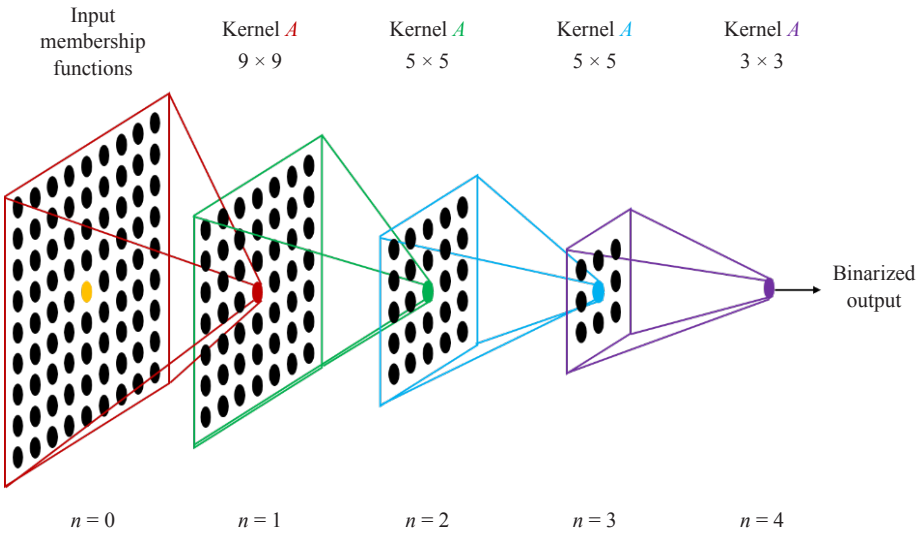

Figure 9. Kernel size of the proposed DTCNN. The inputs are the membership grades of the strong fuzzy classifier. The size of the kernel is reduced in each iteration

The kernels $A(h, j)$ and $B(h, j)$ are based on the model proposed in [23], where a *Gaussian* median filter was used in the definition of the kernel coefficients $A(h, j)$. In our implementation, we proposed a strategy where the size of the kernel will change in a decreasing way. The initial dimension is 9 × 9 and finishes with a 3 × 3 size. With this approach, it is possible to eliminate blobs of false positive regions in a precise way without affecting the definition of the dynamic object. The kernel $B(h, j)$ and the polarization of neuron Z were defined with values equal to zero. Figure 9 shows a diagram of the DTCNN architecture. The initial kernel has a size of 9 × 9. After one iteration, it is reduced to 7 × 7. Then, the kernel change to a size equal to 5 × 5 and finishes with a 3 × 3 size.

The kernel coefficients of $A(h, j)$ are described in Eq. (18)

$$
\begin{aligned}
&V(1) = \begin{bmatrix} 1 & 8 & 28 & 56 & 70 & 56 & 28 & 8 & 1 \end{bmatrix} / 256 \\
&A(1) = 2V(1)^T V(1) \\
&V(2) = \begin{bmatrix} 1 & 6 & 15 & 20 & 15 & 6 & 1 \end{bmatrix} / 64 \\
&A(2) = 2V(2)^T V(2) \\
&V(3) = \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \end{bmatrix} / 16 \\
&A(3) = 2V(3)^T V(3) \\
&V(4) = \begin{bmatrix} 1 & 2 & 1 \end{bmatrix} / 4 \\
&A(4) = 2V(4)^T V(4)
\end{aligned} \tag{18}
$$

The membership values of the high uncertainty level obtained in Eq. (19) are analyzed in conjunction with the DTCNN output I_O.

$$
I_A(x,y,t) = \begin{cases} 1 & \text{If } I_O(x,y,t) = 1 \text{ and } \mu_{Uhigh}(x,y,t) > Th_1 \\ 0 & \text{If } I_O(x,y,t) = 0 \text{ and } \mu_{Uhigh}(x,y,t) > Th_1 \end{cases} \tag{19}
$$

All pixels with a membership grade to U_{high} greater than Th_1 are modified according to the output $I_O(x, y, t)$ of the DTCNN. This modification considers neighbor information to eliminate the mentioned uncertainty. The value of Th_1 was set to 0.9 by experimentation.

2.6 *Background model update*

Once the pixels are classified as foreground or background, it is necessary to update the background models. A Sugeno Fuzzy System calculates the adaptive learning rates of the background models. For the color background model, the fuzzy rules are described below. The constant increments for the learning rates were defined by experimentation.

$$
\begin{aligned}
&\text{If } CD \text{ is } CD_S \text{ Then } LR_C(x,y,t) \text{ is } 0.01 + \Delta_C \\
&\text{If } CD \text{ is } CD_M \text{ Then } LR_C(x,y,t) \text{ is } 0.02 + \Delta_C \quad \forall I_A(x,y,t) = 0 \\
&\text{If } CD \text{ is } CD_B \text{ Then } LR_C(x,y,t) \text{ is } 0.05 + \Delta_C
\end{aligned} \tag{20}
$$

The learning rate associated with the color background model considers illumination changes with the parameter Δ_C. With this strategy, the illumination changes will be included in the color background models reducing the identification of false positive pixels.

The depth background model is updated considering the following fuzzy rules

$$
\begin{aligned}
&\text{If } DD \text{ is } DD_S \text{ and } I_{DD}(x,y,t) < 0 \text{ Then } LR_D(x,y,t) \text{ is } 0.001 \\
&\text{If } DD \text{ is } DD_S \text{ and } I_{DD}(x,y,t) \geq 0 \text{ Then } LR_D(x,y,t) \text{ is } 0.03 \left| I_{DD}(x,y,t) \right| \\
&\text{If } DD \text{ is } DD_M \text{ Then } LR_D(x,y,t) \text{ is } 0.02 \\
&\text{If } DD \text{ is } DD_B \text{ Then } LR_D(x,y,t) \text{ is } 0.01 \\
&\forall \left(I_A(x,y,t) = 0 \wedge CompletePixel \right)
\end{aligned} \tag{21}
$$

The expression (22) allows adding measurements of background pixels to the background if the pixel meets the

condition of *NewPixel*. It can also eliminate pixels with the condition of *EmptyPixel* if they have no measurements in the current frame for a long period.

$$LR_{D0/1}(x,y,t)=\begin{cases}1 & \text{If } I_A(x,y,t)\,\mu_{DS_H}(x,y,t)>Th_D \quad \forall NewPixel\ OR\\ & \text{If } P_{N\_nmd}(x,y,t)>Th_D \qquad\qquad \forall EmptyPixel\\ 0 & \text{otherwise}\end{cases} \tag{22}$$

where Th_D is used to determine which pixels will be added or eliminated from the background model. Th_D generates an α-cut, so that all pixels with a membership to U_{DS_H} higher than Th_D are added to the background. The pixels with a membership to P_{N_nmd} higher than Th_D are eliminated. The value of Th_D was set to 0.95 and was determined by experimentation. Here, the presence of no-measured depth pixels P_{nmd} is defined by

$$P_{nmd}(x,y,t)=P_{nmd}(x,y,t-1)(1-\alpha_{nmd})+\beta_{nmd} \text{ if } \left(I_D(x,y,t)=0\right) \text{ and } \left(I_D(x,y,t-1)=0\right) \tag{23}$$

P_{nmd} will increment every time a pixel with missing depth information is detected in the current frame $I_D(x, y, t) = 0$ and previous frame $I_D(x, y, t - 1) = 0$. α_{nmd}, and β_{nmd} describe the decay and learning rate parameters and were defined with a value of 0.1 and 1, respectively. The P_{nmd} value is also normalized by the rate between β_{nmd} and α_{nmd} in order to define an interval between [0, 1].

Thus, a semi-updated background model for the color and depth map is defined as follows

$$I_{CBS}(x,y,t)=I_{CB}(x,y,t)+LR_C(x,y,t)\left(I_C(x,y,t)-I_{CB}(x,y,t)\right)$$

$$I_{DBS}(x,y,t)=I_{DB}(x,y,t)+\left(LR_D(x,y,t)+LR_{D0/1}(x,y,t)\right)I_{DD}(x,y,t) \tag{24}$$

The decreasing parameters for the color model, α_C, and depth model, α_P, are expressed as

$$\alpha_C(t)=\max\left(1/t,\ 0.005\right)$$

$$\alpha_P(t)=0.1*Sigmoidal\left(\mu_{DDmin}(t),\ 250,\ 0\right) \tag{25}$$

where $\mu_{DD\min}(t)=\min\left(\mu_{DD\min}(t-1),\ \mu_{DD}(t)\right)$, $\mu_{DD}(t)=\dfrac{1}{N_{CP}}\sum_{\forall CompletePixel} I_{DD}(x,y,t)$ and N_{CP} stands for the number of complete pixels. The parameter α_C, which decreases as a function of time with a minimum value of 0.005 and α_P, has a relation with the mean of the depth difference pixels classified as complete pixels.

Finally, the color background model I_{CB} and depth background model I_{DB} are updated as described below

$$I_{CB}(x,y,t+1)=\alpha_C I_C(x,y,t)+(1-\alpha_C)I_{CBS}(x,y,t)\forall\left(ColorPixel\vee EmptyPixel\right)$$

$$I_{DB}(x,y,t+1)=\alpha_D I_D(x,y,t)+(1-\alpha_D)I_{DBS}(x,y,t)\forall CompletePixel \tag{26}$$

In summary, this model has the capability to adapt to global illumination change based on the entropy analysis. Additionally, it absorbs at a fast rate the false positive detection caused by bootstrapping issues. In addition, the update

rules allow the deep background model to include or remove deep measurements based on the identification of missing depth pixels and the similitude between the current frame and the background model. The algorithm to reduce the uncertainty increments the separation of classes by considering the strength of the fuzzy system and by analyzing the color and depth differences. Including the DTCNN with a changing size definition reduces the false positive detections without affecting the foreground result in a precise way. Finally, the fuzzy strategy implemented to update the background models is able to adapt in a precise way to the changes in the scene.

3. Evaluation and results with depth databases

The proposed method was evaluated with the dataset SBM-RGBD [14-15]. This dataset includes 33 videos (~15,000 frames) captured by the Microsoft Kinect sensor. The videos come from a collection of different public datasets: the GSM dataset [8], MULTIVISION [16] Princeton Tracking Benchmark [17], RGB-D object detection dataset [11], and UR Fall Detection Dataset [10]. The videos have a spatial resolution of 640 × 480, and the depth maps vary from 16 or 8 bits. These 33 videos were classified into seven categories: illumination changes, color camouflage, depth camouflage, intermittent motion, out-of-depth sensor range, shadows, and bootstrapping. The metrics used in the SBM-RGBD dataset to measure the performance of each algorithm are Recall (Re), Specificity (Sp), False Positive Rate (FPR), False Negative Rate (FNR), Percentage of Wrong Classification (PWC), Precision (Pr) and F-Measure. Table 1 shows the average metrics results achieved by the FN-DTCNM method and 7 State of the Art algorithms. The ranking place of each method is shown next to the results of each metric. The last column shows the Average Rank result obtained by averaging the ranking of each algorithm on the seven metrics. Our method was implemented in Matlab2017 ® using a Dell XPS 8910, Intel i5-6,400 with 8 GB of RAM. The average computation time was 6 fps.

The RGBD-SOBS algorithm [18] was ranked in the first place. Kim and SCAD [20] methods obtained the second place and the FN-DTCNM method achieved the third one.

Table 1. Comparison of performance metrics using the dataset SBM-RGBD

Method Name	Re		Sp		FPR		FNR		PWC		Pr		F Measure		Avg Rank	Rank
RGBD-SOBS [18]	0.839	3	0.9958	1	0.004	1	0.09	3	1.082	3	0.8796	1	0.8557	3	2.14	1
RGB-SOBS [24]	0.771	6	0.9708	8	0.029	8	0.158	6	5.401	8	0.7247	8	0.7068	8	7.43	7
SRPCA [19]	0.779	5	0.9739	7	0.026	7	0.15	5	3.191	7	0.7474	7	0.7472	5	6.14	5
AvgM-D*	0.707	8	0.9869	5	0.013	5	0.222	8	2.884	6	0.7498	6	0.7157	7	6.43	6
Kim*	0.849	2	0.9947	3	0.005	3	0.079	2	1.029	2	0.8764	2	0.8606	2	2.29	2
SCAD [20]	0.885	1	0.9932	4	0.007	4	0.044	1	0.908	1	0.8698	4	0.8757	1	2.29	2
cwisardH+ [21]	0.762	7	0.9817	6	0.018	6	0.166	7	2.880	5	0.7556	5	0.747	6	6.00	4
FN-DTCNM	0.8373	4	0.9955	2	0.0045	2	0.091	4	1.192	4	0.8751	3	0.8512	4	3.29	3

+The evaluations were achieved in the web page of the dataset, http://rgbd2017.na.icar.cnr.it/SBM-RGBDdataset.html
*The papers of the AvgM-D and Kim methods are not available

RGBD-SOBS defines two background models for color and depth information based on a self-organizing neural map. The color and depth foreground definitions are combined to produce the final segmentation result. The LabGen algorithm [25] runs over the first 100 frames and is considered to initialize the color background model. The

initialization of the depth model is obtained by accumulating the highest depth pixel value of the first 100 frames. Then, a color and depth background subtraction process and a neural map update operation are performed at each time step. Maddalena and Petrosino reported an average processing time of 4 frames per second (fps) for a 640 × 480 resolution. By analyzing the results of Table 1, the maximum difference regarding RGBD-SOBS and FN-DTCNM is in the metric PWC with a difference of 0.1101.

The Simple Combination of the Appearance and Depth information (SCAD) method is based on calculating two likelihoods backgrounds of depth and appearance in combination with graph cuts [20]. The likelihood of the depth background uses the distance between the input depth frame and the background model and the classification of the depth pixel (measure of *nmd* pixel; constant *nmd*, rippling *nmd*, or constantly measured pixel). For the appearance, a background model scale-invariant local ternary pattern (SILTP) is computed as a texture-base feature in combination with the background subtraction algorithm ViBe. The analysis of the depth and appearance information is performed using a graph-cut approach. As a result, the foreground definitions that they reported are very accurate but at the cost of an execution time of 1.95 fps.

cwisardH+ [21] implements two different weightless neural networks to model the pixel RBG color and depth. These networks are separately processed. The Region of Interest (ROI) information provided in the SBM-RGBD video dataset is used by cwisardH+ to restrict background learning in these areas. The outputs of both models are post-processed by erosion and dilation filters and combined with the OR operator. cwisardH+ reports a running time of 8 fps in training and 2 fps in classification with 720 × 480 resolution videos.

SRPCA [19] is a semi-online algorithm that implements a dense optical flow algorithm to define an initial motion mask. Then, spatiotemporal graph Laplacians encode the local similarity in the dynamic sequence. Finally, an objective function is solved by a matrix factorization and a minimization strategy. The authors reported a running time of ~22 sec for processing 240 × 320 × 90 video data.

The performance of the algorithms considering the 7 categories of videos is illustrated in Table 2.

In this previous analysis, considering the performance by video category, it can be observed from Table 2 that the proposed method FN-DTCNM achieved first place in three categories: Illumination Change, Intermittent Motion, and Out-of-Range. Furthermore, FN-DTCNM obtained second place with the Depth Camouflage videos and third place with the Shadows videos. These results demonstrate the robustness of the FN-DTCNM model. The automatic displacement, ΔC, defined in Eq. (5), considering the *Color difference* fuzzy variable, the weighting parameter, and the learning rate update, results in a very good strategy to detect illumination changes automatically. In addition, the treatment of no-measured depth *nmd* pixels, uncertainty pixel classification, and the analysis of the strong fuzzy classifier, improved the results with the videos of the Out of Range category. In addition, because our model could accurately combine the color and depth information in the foreground classification, the possible errors caused by abandoned or removed foreground objects were minimal, as demonstrated by the results of the Intermittent Motion category.

Maddalena and Petrosino reported in [18] some qualitative results of the RGBD-SOBS algorithm and we present them in Figure 10. The videos correspond to *BootStrapping_ds* (frame 208), *shadows2* (frame 243), *Chair-Box* (frame 350), and *DCamSeq2* (frame 420). The last two columns present the foreground definition of the RGBD-SOBS and FN-DTCNM algorithms. In the first video, RBGD-SOBS reported false positive detections, and FN-DTCNM achieved a better definition of the dynamic object. The second video is the opposite. The false positive detections were produced with FN-DTCNM. In the third video, FN-DTCNM achieved a better true positive detection, and the same situation is reported in the last video.

Table 2. Average results in each one of the 7 video categories

Method Name	Re		Sp		FPR		FNR		PWC		Pr		F Measure		Avg Rank	Rank
							Bootstrapping									
RGBD-SOBS	0.884	2	0.9925	4	0.008	4	0.116	2	2.327	3	0.908	5	0.8917	3	3.29	3
RGB-SOBS	0.802	4	0.9814	7	0.019	7	0.198	4	4.4221	6	0.8165	6	0.8007	6	5.71	6
SRPCA	0.728	5	0.9914	5	0.009	5	0.272	5	3.7409	5	0.9164	4	0.8098	5	4.86	5
AvgM-D	0.459	8	0.9861	6	0.014	6	0.541	8	7.196	7	0.6941	7	0.535	8	7.14	7
Kim	0.881	3	0.9965	2	0.004	2	0.12	3	1.5227	1	0.9566	1	0.9169	1	1.86	1
SCAD	0.9	1	0.994	3	0.006	3	0.1	1	1.8015	2	0.9319	3	0.9134	2	2.14	2
cwisardH+	0.573	7	0.9616	8	0.038	8	0.427	7	8.1381	8	0.5787	8	0.5669	7	7.57	8
FN-DTCNM	0.7270	6	0.9970	1	0.0030	1	0.2730	6	3.7335	4	0.9441	2	0.8145	4	3.43	4
							ColorCamouflage									
RGBD-SOBS	0.956	4	0.9927	1	0.007	1	0.044	4	1.2161	5	0.9434	4	0.9488	4	3.29	3
RGB-SOBS	0.431	8	0.9767	7	0.023	7	0.569	8	16.04	8	0.8018	8	0.4864	8	7.71	8
SRPCA	0.848	7	0.9389	8	0.061	8	0.152	7	4.3124	7	0.8367	6	0.8329	7	7.14	7
AvgM-D	0.9	6	0.9793	6	0.021	6	0.1	6	2.0719	6	0.8096	7	0.8508	6	6.14	6
Kim	0.974	2	0.9927	1	0.007	1	0.026	2	0.7389	2	0.9754	1	0.9745	2	1.57	1
SCAD	0.988	1	0.9904	4	0.01	4	0.013	1	0.7037	1	0.9677	2	0.9775	1	2.00	2
cwisardH+	0.953	5	0.9849	5	0.015	5	0.047	5	1.1931	4	0.9502	3	0.951	3	4.29	5
FN-DTCNM	0.9713	3	0.9907	3	0.0093	3	0.0287	3	1.0283	3	0.9246	5	0.9453	5	3.57	4
							DepthCamouflage									
RGBD-SOBS	0.84	6	0.9985	1	0.002	1	0.16	6	0.9778	3	0.9682	1	0.8936	4	3.14	3
RGB-SOBS	0.973	2	0.9856	7	0.014	7	0.028	2	1.5809	5	0.8354	7	0.8935	5	5.00	5
SRPCA	0.868	5	0.9778	8	0.022	8	0.132	5	2.9944	8	0.785	8	0.8083	7	7.00	8
AvgM-D	0.837	7	0.9922	6	0.008	6	0.163	7	1.6943	6	0.886	6	0.8538	6	6.29	6
Kim	0.87	4	0.9968	3	0.003	3	0.13	4	0.982	4	0.9433	4	0.9009	3	3.57	4
SCAD	0.984	1	0.9963	4	0.004	4	0.016	1	0.4432	1	0.9447	3	0.9638	1	2.14	1
cwisardH+	0.682	8	0.9949	5	0.005	5	0.318	8	2.4049	7	0.9016	5	0.7648	8	6.57	7
FN-DTCNM	0.8939	3	0.9974	2	0.0026	2	0.1061	3	0.7762	2	0.9527	2	0.9186	2	2.29	2
							IlluminationChanges									
RGBD-SOBS	0.451	5	0.9955	1	0.005	1	0.049	5	0.9321	2	0.4737	2	0.4597	3	2.71	2
RGB-SOBS	0.437	7	0.9715	8	0.029	8	0.063	7	3.5022	8	0.4759	1	0.4527	5	6.29	7
SRPCA	0.48	1	0.9816	7	0.018	7	0.021	1	1.9171	6	0.4159	8	0.4454	7	5.29	6
AvgM-D	0.339	8	0.9858	6	0.014	6	0.161	8	3.0717	7	0.4188	7	0.3569	8	7.14	8
Kim	0.448	6	0.9935	3	0.007	3	0.052	6	1.1395	5	0.4587	4	0.4499	6	4.71	5
SCAD	0.47	3	0.9927	4	0.007	4	0.03	3	0.9715	3	0.4567	5	0.461	2	3.43	3
cwisardH+	0.471	2	0.9914	5	0.009	5	0.029	2	1.0754	4	0.4504	6	0.4581	4	4.00	4
FN-DTCNM	0.4683	4	0.9954	2	0.0046	2	0.0317	4	0.7533	1	0.4709	3	0.4681	1	2.43	1

Table 2. (cont.)

Method Name	Re		Sp		FPR		FNR		PWC		Pr		F Measure		Avg Rank	Rank
	IntermittentMotion															
RGBD-SOBS	0.892	6	0.997	1	0.003	1	0.108	6	0.8648	3	0.9544	1	0.9202	4	3.14	4
RGB-SOBS	0.927	3	0.9028	8	0.097	8	0.074	3	9.3877	8	0.4054	8	0.5397	8	6.57	7
SRPCA	0.889	7	0.9629	6	0.037	6	0.111	7	3.7026	6	0.7208	6	0.7735	6	6.29	6
AvgM-D	0.898	5	0.9912	5	0.009	5	0.102	5	1.4603	5	0.9115	5	0.9027	5	5.00	5
Kim	0.942	2	0.9938	3	0.006	3	0.058	2	0.9213	4	0.9385	3	0.939	1	2.57	1
SCAD	0.956	1	0.9914	4	0.009	4	0.044	1	0.8616	2	0.9243	4	0.9375	2	2.57	1
cwisardH+	0.809	8	0.9558	7	0.044	7	0.191	8	5.0851	7	0.5984	7	0.6633	7	7.29	8
FN-DTCNM	0.9257	4	0.9957	2	0.0043	2	0.0743	4	0.6205	1	0.9475	2	0.9346	3	2.57	1
	OutOfRange															
RGB-SOBS	0.89	6	0.9896	6	0.01	6	0.11	6	1.361	6	0.8237	6	0.8527	6	6.00	6
SRPCA	0.879	7	0.9878	7	0.012	7	0.122	7	1.61	7	0.7443	7	0.8011	7	7.00	7
AvgM-D	0.632	8	0.986	8	0.014	8	0.368	8	2.7663	8	0.636	8	0.6325	8	8.00	8
Kim	0.904	4	0.9961	4	0.004	4	0.096	4	0.8228	4	0.9216	4	0.912	4	4.00	4
SCAD	0.929	2	0.9965	3	0.004	3	0.071	2	0.5711	2	0.9357	2	0.9309	2	2.29	3
cwisardH+	0.896	5	0.9956	5	0.004	5	0.104	5	0.8731	5	0.9038	5	0.8987	5	5.00	5
FN-DTCNM	0.9431	1	0.9970	2	0.0030	2	0.0569	1	0.5934	3	0.9290	3	0.9355	1	1.86	1
	Shadows															
RGBD-SOBS	0.932	4	0.997	1	0.003	1	0.068	4	0.7001	1	0.9733	1	0.95	1	1.86	1
RGB-SOBS	0.936	3	0.9881	5	0.012	5	0.064	3	1.5128	6	0.914	5	0.9218	6	4.71	6
SRPCA	0.759	8	0.9768	8	0.023	8	0.241	8	4.0602	8	0.8128	8	0.7591	8	8.00	8
AvgM-D	0.881	7	0.9876	7	0.012	7	0.119	7	1.933	7	0.8927	7	0.8784	7	7.00	7
Kim	0.927	6	0.9934	3	0.007	3	0.073	6	1.0771	4	0.9404	3	0.9314	4	4.14	4
SCAD	0.967	1	0.991	4	0.009	4	0.034	1	1.0093	3	0.9276	4	0.9458	2	2.71	2
cwisardH+	0.952	2	0.9877	6	0.012	6	0.048	2	1.3942	5	0.9062	6	0.9264	5	4.57	5
FN-DTCNM	0.9320	5	0.9954	2	0.0046	2	0.0680	5	0.8455	2	0.9568	2	0.9419	3	3.00	3

In order to perform a deeper comparison of qualitative segmentation results, we searched state-of-the-art models in different papers that reported their segmentation results. These methods have not been evaluated in the SBM-RGBD dataset; therefore, they were not included in the previous Tables. Figure 11 shows a qualitative comparison between the FN-DTCNM and the *Depth-Extended Codebook* DECB model [16]. The first row corresponds to frame 286 of the *ChairBox* video. Natural illumination changes and many pixels with missing depth information mainly affect the foreground detection. A visual comparison of both methods shows that DECB produced more FP detections. The second row shows the segmentation results with the *Hallway* video in frame 258. Color camouflage is one of the main issues that affect the identification of a white package (marked with a red circle). Even when DECB identified this package, it produced many FP pixels because of shadows. Contrary, the result with FN-DTCNM is very accurate. The segmentation results produced with the *Shelves* video in frame 364 are shown in the third row. This video sequence has slight illumination changes, shadows, and color camouflage. The identification of the object enclosed within a red

circle is complicated because of the color camouflage. Even when the identification of this object is not complete in FN-DTCNM, its definition is much better compared with DECB. In addition, DECB has many FP detections in the foreground result.

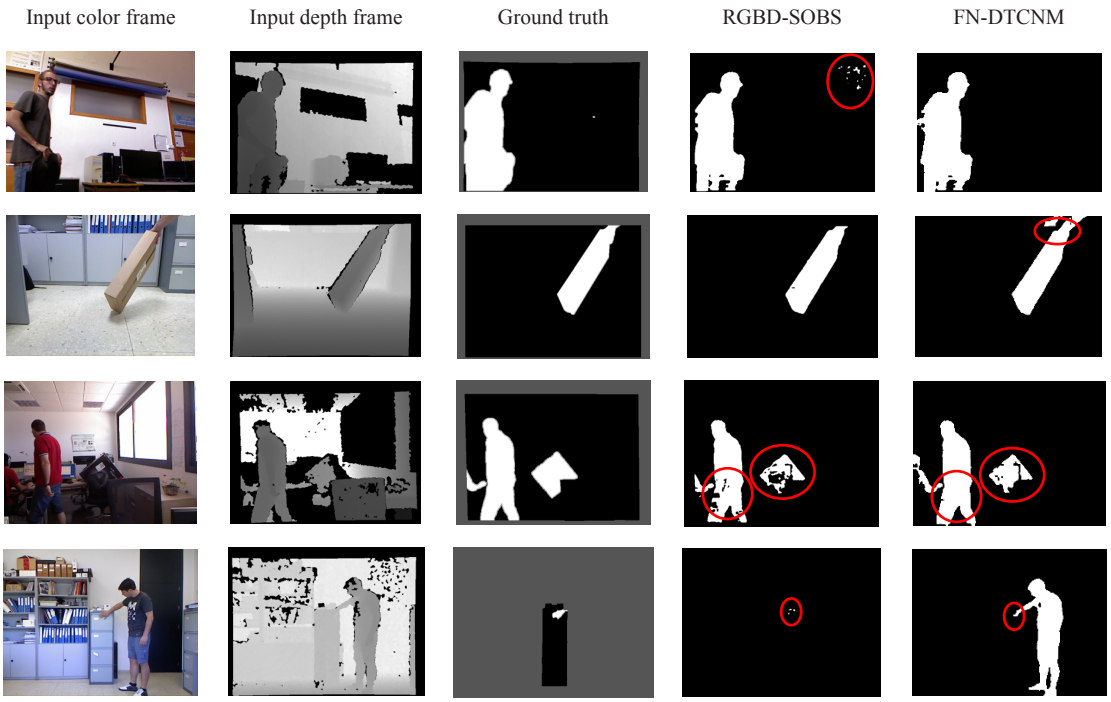

Figure 10. Qualitative segmentation results of the RGBD-SOBS [18] and FN-DTCNM models. The videos correspond to *BootStrapping_ds* (frame 208), *shadows2* (frame 243), *Chair-Box* (frame 350), and *DCamSeq2* (frame 420)

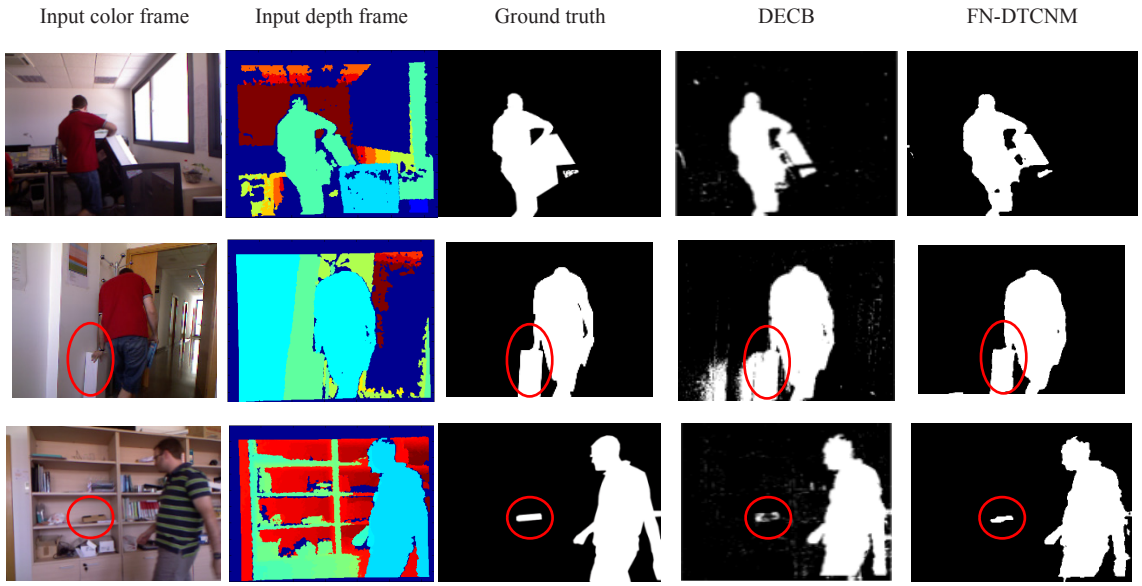

Figure 11. Qualitative segmentation results of the DECB [16] and FN-DTCNM models. The first row presents the results with the *ChairBox* video at frame 286. The second row shows the *Hallway* video at frame 258. The third row presents the *Shelves* video at frame 364

Figure 12 shows segmentation results reported in [11] and [26] with the CL_W and MoG-RegPRE models. The first row shows the *GenSeq* sequence at frame 984. This video has issues related to shadows of moving objects, color camouflage, and noisy depth data. The foreground identification reported with the CL_W and MoG-RegPRE models has many FP detections. The result obtained with the FN-DTCNM is more precise. The second row presents the *ShSeq* video at frame 445. This sequence presents hard shadows produced by the dynamic object. The result of CL_W has many FP. MoG-RegPRE and FN-DTCNM obtained a more accurate detection of the foreground.

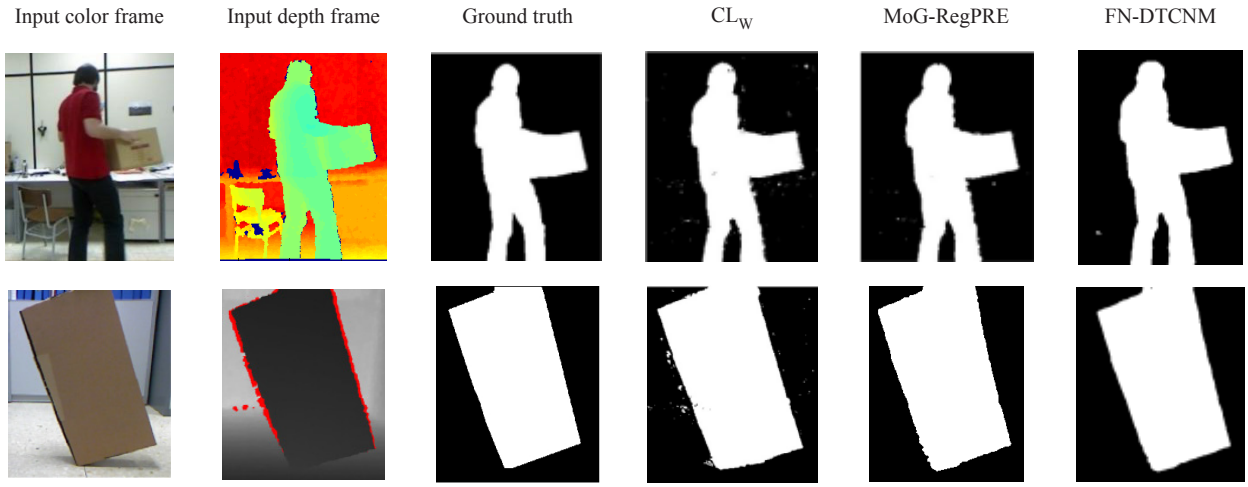

Figure 12. Qualitative segmentation results of the CL_W [11], MoG-RegPRE [26], and FN-DTCNM models. The first row presents the results with the *GenSeq* video at frame 984. The second row shows the *ShSeq* video at frame 445

4. Conclusions

This paper proposed a fuzzy neural classifier that considers color and depth information to define the background in video sequences in order to find dynamic objects. The model includes a weak fuzzy classifier that performs an initial foreground and background separation based on color and depth differences between the actual frame and background models. The outputs of this fuzzy system are weighted according to the result of the color and depth noise modeling. The proposed method defines dynamic objects more accurately by analyzing a degree of uncertainty and the strength of decisions in combination with the weighting results. This analysis is performed with a strong fuzzy classifier. The final stage of the foreground detection is implemented with a Discrete-Time Cellular Neural Network (DTCNN) to improve the foreground definition. This network considers the membership grades of color and depth differences of the pixel under analysis and its neighborhood to eliminate false positive detections. Once the pixels are classified, the color and depth background models are updated based on a fuzzy learning rate strategy. To demonstrate the robustness of the proposed method, it was evaluated quantitatively with the new SBM-RGBD database and qualitatively against state-of-the-art methods achieving very competitive results. Our proposal obtains the first place in the Illumination Change, Intermittent Motion, and Out-of-Range categories of the SBM-RGBD database, second place in the Depth Camouflage category, and third place in the Shadow category. A qualitative comparison against the best method ranked in the SBM-RGBD database shows that our results are not very different. In some cases, our method produced a better definition of the dynamic object with a reduced false positive detection. Additionally, our method has the best processing time reported at 6 fps. Therefore, the FN-DTCNM method can be contemplated as a good and new alternative for dynamic object detection using RGBD information. Although the proposed method showed comparable and better results than state-of-the-art methods, some points may be considered for future work. The main weaknesses found in the proposed method are related to color camouflage and shadows. Therefore, the FN-DTCNM needs to be improved to face those situations better. Another important aspect that should be considered for improvement is the frame processing speed. Notwithstanding that FN-DTCNM has a good processing time, it must be improved to be used at the RGBD sensor's

speeds.

Acknowledgements

The authors wish to extend their thanks to Tecnologico Nacional de Mexico/I. T. Chihuahua for the support provided to carry out this work, under grants 5162.19-P, and 10071.21-P. We also thank the peer reviewers for their valuable suggestions and time invested in reviewing our manuscript.

Conflict of interest

The authors declare no competing financial interest.

References

[1] Zhou DF, Fremont V, Quost B, Dai YC, Li HD. Moving object detection and segmentation in urban environments from a moving platform. *Image and Vision Computing*. 2017; 68: 76-87.
[2] Xue HY, Liu Y, Cai D, He XF. Tracking people in RGBD videos using deep learning and motion clues. *Neurocomputing*. 2016; 204: 70-76.
[3] Adama DA, Lotfi A, Langensiepen C, Lee K, Trindade P. Human activity learning for assistive robotics using a classifier ensemble. *Soft Computing*. 2018; 22(21): 7027-7039.
[4] Mohanty SK, Rup S, Swamy M. An improved scheme for multifeature-based foreground detection using challenging conditions. *Digital Signal Processing*. 2021; 113: 103030.
[5] Zhao XY, Wang GL, He ZX, Jiang HL. A survey of moving object detection methods: A practical perspective. *Neurocomputing*. 2022; 503: 28-48.
[6] Garcia-Garcia A, Orts-Escolano S, Oprea S, Villena-Martinez V, Martinez-Gonzalez P, Garcia-Rodriguez J. A survey on deep learning techniques for image and video semantic segmentation. *Applied Soft Computing*. 2018; 70(1): 41-65.
[7] Lu M, Chen H, Lu P. Perception and avoidance of multiple small fast moving objects for quadrotors with only low-cost RGBD camera. *IEEE Robotics and Automation Letters*. 2022; 7(4): 11657-11664.
[8] Moya-Alcover G, Elgammal A, Jaume-i-Capo A, Varona J. Modeling depth for nonparametric foreground segmentation using RGBD devices. *Pattern Recognition Letters*. 2017; 96: 76-85.
[9] Trabelsi R, Jabri I, Smach F, Bouallegue A. Efficient and fast multi-modal foreground-background segmentation using RGBD data. *Pattern Recognition Letters*. 2017; 97: 13-20.
[10] Kwolek B, Kepski M. Human fall detection on embedded platform using depth maps and wireless accelerometer. *Computer Methods and Programs in Biomedicine*. 2014; 117(3): 489-501.
[11] Camplani M, Salgado L. Background foreground segmentation with RGB-D Kinect data: An efficient combination of classifiers. *Journal of Visual Communication and Image Representation*. 2014; 25(1): 122-136.
[12] Sultana M, Bouwmans T, Giraldo JH, Jung SK. Robust foreground segmentation in RGBD data from complex scenes using adversarial networks. *Communications in Computer and Information Science*. 2021; 1405: 3-16.
[13] Janus P, Kryjak T, Gorgon M. Foreground object segmentation in RGB-D data implemented on GPU. In: Bartoszewicz A, Kabziński J, Kacprzyk J. (eds.) *Advanced, Contemporary Control*. Springer, Cham; 2020. p.809-820.
[14] Camplani M, Maddalena L, Moyá AG, Petrosino A, Salgado L. New trends in image analysis and processing-ICIAP 2017 workshops. In: Cham S, Battiato G, Gallo GM, Leo FM. (eds.) *Lecture Notes in Computer Science*. Springer International Publishing; 2017.
[15] Camplani M, Maddalena L, Moya AG, Petrosino A, Salgado L. *SBM-RGBD Dataset*. 2017. Available from: http://rgbd2017.na.icar.cnr.it/SBM-RGBDdataset.html [Accessed 4th July 2022].
[16] Fernandez-Sanchez EJ, Diaz J, Ros E. Background subtraction based on color and depth using active sensors. *Sensors*. 2013; 13(7): 8895-8915.
[17] Song S, Xiao J. Tracking revisited using RGBD camera: Unified benchmark and baselines. *2013 IEEE*

International Conference on Computer Vision. Sydney, NSW, Australia; 2013.
[18] Maddalena L, Petrosino A. Exploiting color and depth for background subtraction. *New Trends in Image Analysis and Processing-ICIAP 2017*. Springer, Cham; 2017. p.254-265.
[19] Javed S, Bouwmans T, Sultana M, Jung S. Moving object detection on RGB-D videos using graph regularized spatiotemporal RPCA. *New Trends in Image Analysis and Processing-ICIAP 2017*. Springer, Cham; 2017. p.230-241.
[20] Minematsu T, Shimada A, Uchiyama H, Taniguchi R-i. Simple combination of appearance and depth for foreground segmentation. *New Trends in Image Analysis and Processing-ICIAP 2017*. Springer, Cham; 2017. p.266-277.
[21] De Gregorio M, Giordano M. CwisarDH+: Background detection in RGBD videos by learning of weightless neural networks. *New Trends in Image Analysis and Processing-ICIAP 2017*. Springer, Cham; 2017. p.242-253.
[22] Nguyen CV, Izadi S, Lovell D. Modeling kinect sensor noise for improved 3D reconstruction and tracking. *2012 Second International Conference on 3D Imaging, Modeling, Processing, Visualization Transmission*. Zurich, Switzerland; 2012.
[23] Ramirez-Quintana JA, Chacon-Murguia MI. Self-adaptive SOM-CNN neural system for dynamic object detection in normal and complex scenarios. *Pattern Recognition*. 2015; 48(4): 1137-1149.
[24] Maddalena L, Petrosino A. The SOBS algorithm: What are the limits? *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*. Providence, RI, USA; 2012.
[25] Laugraud B, Piérard S, Braham M, Van Droogenbroeck M. Simple median-Based method for stationary background generation using background subtraction algorithms. *New Trends in Image Analysis and Processing-ICIAP 2015 Workshops*. Springer, Cham; 2015. p.477-484.
[26] Camplani M, del Blanco CR, Salgado L, Jaureguizar F, Garcia N. Multi-sensor background subtraction by fusing multiple region-based probabilistic classifiers. *Pattern Recognition Letters*. 2014; 50: 23-33.