

Small Target Detection in Night-Time Videos Using Persistence Filter

Anonymous CVPR submission

Paper ID ****

Abstract

This paper proposes a reliable method for detecting small targets in noisy and/or low-contrast videos. Examples of this kind are night-time videos in low-lighting conditions. Traditional background subtraction methods which rely on the difference between the image and the background model suffer from sensitivity to a set of difference thresholds which result in either a high rate of false alarms or high rate of target miss detection. Our method tracks objects in the scene and models their persistence over time using a probabilistic model called a Persistence Filter. An adaptive detection threshold is selected for each object based on the global noise level of the scene as well as properties of that object including area, contrast and speed. Experimental results show the effectiveness of this algorithm especially in low-contrast and noisy situations where classical background subtraction methods fail.

1. Introduction

One of the primary stages in an intelligent visual tracking system is background subtraction. In this stage, the colored or gray-level image of the current frame acquired by the camera is compared against a background model and the regions which show significant difference with the model

are selected as foreground regions. The background model consequently is adapted to address the gradual changes in the scene. Most of the approaches use a set of probabilistic cues such as the pixel value mean, standard deviation and distribution to model the background and corresponding thresholds are deployed to determine which pixels or regions in the scene are part of the foreground and which ones are not. Here the challenge is to determine suitable thresholds so that all desired targets are detected and all the noise objects are filtered. This becomes harder as the noise level of the scene increases or the contrast of the targets (*i.e.* the distance of the color or gray level pixel values to the background) is comparable to the noise level. In such scenarios, selecting small values for the thresholds results in a large number of noises so called false alarms being detected as targets. In contrast, large thresholds prevent the desired targets of low-contrast to be detected. This papers addresses this problem by retaining a map which stores the history of presence of each object over time. The underlying idea is that real targets move smoothly over time therefore they preserve their presence in a local neighborhood in a small temporal window of frames. In contrast appearance of noises in random positions shows incoherency of their motions. On the other hand, real targets show more persistency than the noise objects. In the proposed approach, the

108 detection of the targets is based on filtering the persistence
109 map using a set of adaptive thresholds which are functions
110 of the global noise level of the scene as well as the prop-
111 erties of individual objects. These properties include the
112 area and contrast which can be deployed as cues to distin-
113 guish noise and targets. The presented approach is robust in
114 multiple senses. The shortage in any of the expected object
115 properties does not result in failure of the algorithm, but as
116 explained in the paper, it just makes the detection process
117 lengthier as higher level of persistency from the target is re-
118 quired. Also, temporary miss detection of targets in the pre-
119 liminary background subtraction step does not cause losing
120 the track for a long time. In fact, the persistence filter can
121 retrieve the track as soon as the target reappears.

132 2. Related Work

133 Background subtraction is a well studied area as it is
134 a fundamental need for many vision system. Variety of
135 methods with the goal of optimizing speed, memory and
136 accuracy have been proposed [9]. A naive method com-
137 pares the image with a background image captured in the
138 past and thresholds the difference. A better approach com-
139 puts the background image by computing the average or
140 median of a sequence of images [1]. Statistical methods
141 model the distribution of the pixel values using a running
142 gaussian average [12] or mixture of gaussians [11, 10]. To
143 relax the assumption of gaussian pixel intensity distribution,
144 non-parametric methods are deployed. In [2], kernel density
145 estimation technique is utilized to model the background
146 distribution. Also, in [6], background model is constructed
147 by clustering color and intensity values in terms of a set of
148 codewords.

162 In order to model the background, several different im-
163 age features could be employed. While in most of the al-
164 gorithms, the intensity of individual pixels are used, some
165 others use other features such as gradient [5], motion [7]
166 and texture [4]. In a recent paper, Parag *et al.* [8], employed
167 a boosting algorithm to select the most useful features and
168 discriminate foreground objects from the background.

169 Almost all the the background subtraction algorithms de-
170 pend on a set of global or local thresholds. Therefore, one of
171 the main challenges is to determine these thresholds to min-
172 imize the rate of false alarms and miss detections. Gao *et*
173 *al.* in [3] discussed the error analysis for mixture of gaussian
174 model by trade off of false alarm and miss detection. Con-
175 sidering this trade-off, the problem arises when the noise
176 level increases and becomes comparable to the signal level.
177 This usually happens in dark scenes where signal (pixel)
178 values decrease and the noise level increases simultane-
179 ously. In such a scenario, trying to reduce one of these two
180 parameters, increases the other one significantly.

181 TO BE CONTINUED.

182 3. Noise vs. Target

183 There are two basic types of noise appearing in images.
184 The first type so called as camera/illumination based in-
185 clude noises appear due to either inherent properties of the
186 camera as the acquisition device or the type of illumina-
187 tion of the scene. The former source of noise includes
188 Automatic Gain Control (AGC) and CCD defects, and the
189 latter one includes sudden illumination change and low-
190 illumination effect. (THIS PART MIGHT NEED SOME
191 CORRECTION!)

192 The second type of noise is due to the presence of unde-

sired targets in the scene. For example, waving leaves, footprints on the ground, reflections in the glasses, and shadows are of this type. The common characteristic of all these objects is that they have a behavior similar to the desired targets. Therefore they cannot be eliminated without any recognition-base approach. Therefore, dealing with this category of noises is beyond the scope of this paper. On the other hand, our goal is to eliminate the noises of the first category while retaining the desired targets as the foreground objects. Nevertheless, detecting some of the objects of the second category as targets is inevitable. For the rest of this paper, we simply refer to the noises of the second type as noise objects.

To be able to filter noises while retaining the real targets, we need to focus on the differences in property and behavior of these two categories of objects. The three common differences between a noise and a real target follow:

1. Area: Due to the inherent source of the noises, they are distributed mostly homogenously throughout the entire image. As a result, they are less likely to aggregate and create a large connected component therefore they are usually smaller in size than the real targets.
2. Speed: A real target is coherent in motion in the sense that its positions in consecutive frames have small spatial distance whereas noise objects appear in random places. In fact, for a sample noise pixel at time t , the location of the closest noise pixel at time $t + 1$ is likely to be more than a few pixels away. On the other hand, if we assume that these two pixels are associated with a single moving target, it would be a fast speed one.
3. Contrast: A target usually has higher contrast than

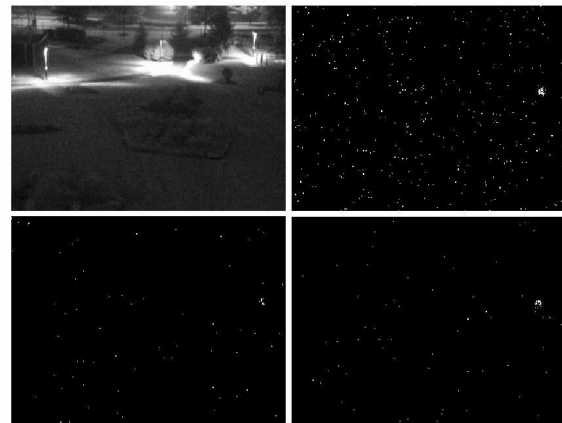


Figure 1. Noise vs. Target: Top-Left: A sample input image. Top-Right: The foreground mask using low threshold. Bottom-Left: The foreground mask using high threshold. Bottom-Right: The foreground mask of the next time instant using high threshold.

a noise in a way that if we create multiple foreground maps using multiple thresholds, the number of noise objects changes significantly between the maps whereas targets are more likely to be present in several maps. This will be discussed in detail in section 7.1.

Figure 1 depicts the three differences of noise and target by showing the foreground mask generated using a classical background subtraction method at two consecutive time instants and two different thresholds. The single real target in the scene is easily detectable using its relatively large size. Meanwhile, the difference between the number of noise objects in the the foreground masks of different thresholds shows that most of the noise objects have low contrast therefore are filtered in the high threshold foreground mask. And finally, comparing the position of the noises in the foreground mask of two consecutive frames (bottom row) shows the motion incoherency of the noises whereas the motion of the target seems to be coherent (*i.e.* insignificant motion in short time interval). We exploit the mentioned differences to detect the targets and eliminate the noises as explained in

the subsequent sections.

4. Overview of the Method

The target detection algorithm presented in this paper include the following steps:

1. Since the method relies on the history of presence of targets over time so called persistency, a real-valued map of the same size as the image named as *Persistence Map* is created to store a signature of this persistency for each individual pixel throughout the time. This map is updated at each time instant as explained in section 5.
2. As the number of noise objects in the scene increases, target detection becomes more challenging and targets are expected to show more persistency or longer history of coherent presence in a local neighborhood. Therefore to determine how much persistency is required, the global noise level of the scene is estimated and a global persistence threshold is selected as discussed in section 6.
3. For each individual objects in the scene, the basic properties including the area and contrast are estimated as explained in section 7.1. This information is used to adjust the persistence threshold so that targets of larger area and/or higher contrast are detected more easily. However, as mentioned earlier, this does not mean that targets of smaller area or lower contrast cannot be detected. In fact, they just need to show more temporal persistency so they can get distinguished from the noises of similar area or contrast.

5. Persistence Map

To keep track of the presence of targets in the video sequence, a real-valued map of the same size as the image is defined. Since the elements of this map show the persistency of objects appearance over time in the neighborhood of corresponding pixels, we name it *Persistence Map*. The structure of this map follows:

Let us define $\mathbf{w}(x, y, t)$ as an n-tuple vector of binary values associated with the pixel $im(x, y, t)$ at position (x, y) and time instant t :

$$\mathbf{w}(x, y, t) = [w_1(x, y, t) \dots w_n(x, y, t)]^T \quad (1)$$

where

$$w_i(x, y, t) = \begin{cases} im(x, y, t) & i = 1 \\ \max(w_{i-1}(x', y', t - 1)) & i \neq 1 \end{cases} \quad (2)$$

with $(x', y') \in N(x, y)$, a neighborhood of the position (x, y) for taking into account the motion of the targets.

Now, the value of the Persistence map at the position (x, y) and time instant t is defined as:

$$P(x, y, t) = \alpha \mathbf{w}(x, y, t)^T \mathbf{c}(d) \quad (3)$$

where the element i of the n-tuple decay vector $\mathbf{c}(d)$ is defined as:

$$c_i(d) = d^{i-1} \quad d \in [0, 1] \quad , \quad 1 \leq i \leq n \quad (4)$$

and the normalizing factor α with the purpose of limiting the range of the persistence map value to the range of 0 and

1 is defined as:

$$\alpha = \frac{1}{\sum_{i=1}^n d^{i-1}} = \frac{1-d}{1-d^n} \quad (5)$$

In fact, the value $P(x, y, t)$ can represent the probability of presence of a target at position (x, y) and time instant t .

Figure 2 illustrates how the persistence map works by showing the map in the right column and the corresponding image in the left column for three consecutive frames. The black cells show where foreground pixels are detected by a classical background subtraction method. The neighborhood $N(x, y)$ for this example is defined as a square of size 3 centered at the targeted pixel.

As it can be seen from the figure, the pixel at $(4, 1, t+2)$ gets the largest value as it is associated with a pixel-sized object (marked with white dots) moving coherently in the three frames.

Updating the persistence map at each frame, we need a threshold to select pixels being recognized as foregrounds. The threshold at each pixel determines whether that pixel has been persistent enough in the past few frames or not. For example, let us assume that a unique threshold denoted as τ is used for all the pixels in the example of figure 2 such that

$$\alpha + \alpha d < \tau \leq \alpha + \alpha d + \alpha d^2 \quad (6)$$

then, the pixel marked with a circle would be the only one to be recognized as foreground.

Even though, a unique threshold τ might work well in some scenarios, it lacks using the properties of individual objects. As explained in section 3, area, contrast and speed are valuable information which can be exploited to distinguish noises and targets. For instance, if a large high-

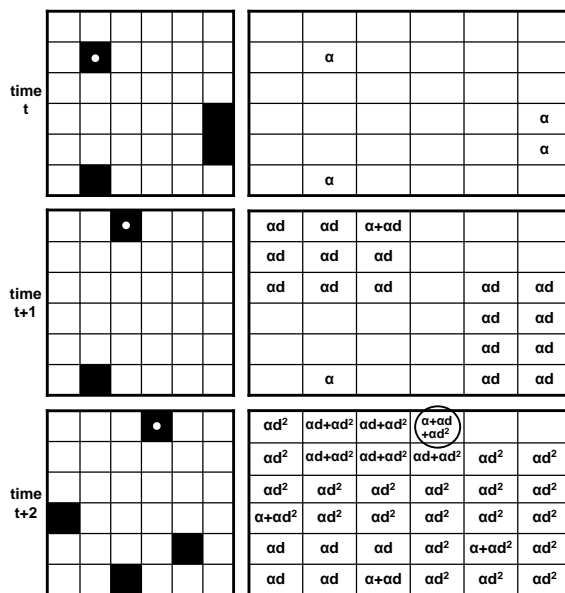


Figure 2. Persistence Map

contrast object is observed, it has a higher chance of being a real target, therefore we can use a lower threshold for evaluating its persistency. In contrast, if a real target has a small area and low contrast, it needs to appear in a local region in the scene for a longer period of time to prove that is not in fact a noise object! In conclusion, the persistence threshold should be an object-based quantity. Sections 6 and 7.2 discuss how to determine this threshold.

5.1. The Effect of the Decay Factor d

The value of the decay factor d , which by definition should be in the range of 0 and 1, indicates how much value we consider for the presence of an object in a local region in the past. The larger the value d is, the more value is given to the past frames. As shown in the table of figure 3, if $d \leq 0.5$, then all the permutations of the vector $\mathbf{w}(x,y,t)$ follow the order of the binary numbers where in terms of significance, each bit dominates all the bits to its right. On the other hand, if an object appears in a particular pixel position

d=0.5		d=0.75		d=1	
$w(x,y,t)$	P	$w(x,y,t)$	P	$w(x,y,t)$	P
0 0 0 0	0.00	0 0 0 0	0.00	0 0 0 0	0.00
0 0 0 1	0.07	0 0 0 1	0.15	0 0 0 1	0.25
0 0 1 0	0.13	0 0 1 0	0.21	0 0 1 0	0.25
0 0 1 1	0.20	0 1 0 0	0.27	0 1 0 0	0.25
0 1 0 0	0.27	0 0 1 1	0.36	1 0 0 0	0.25
0 1 0 1	0.33	1 0 0 0	0.37	0 0 1 1	0.50
0 1 1 0	0.40	0 1 0 1	0.43	0 1 0 1	0.50
0 1 1 1	0.47	0 1 1 0	0.48	0 1 1 0	0.50
1 0 0 0	0.53	1 0 0 1	0.52	1 0 0 0	0.50
1 0 0 1	0.60	1 0 1 0	0.57	1 0 1 0	0.50
1 0 1 0	0.67	0 1 1 1	0.63	1 1 0 0	0.50
1 0 1 1	0.73	1 1 0 0	0.64	0 1 1 1	0.75
1 1 0 0	0.80	1 0 1 1	0.73	1 0 1 1	0.75
1 1 0 1	0.87	1 1 0 1	0.79	1 1 0 1	0.75
1 1 1 0	0.93	1 1 1 0	0.85	1 1 1 0	0.75
1 1 1 1	1.00	1 1 1 1	1.00	1 1 1 1	1.00

Figure 3. The Effect of the Decay Factor on the Order of different permutations of the vector $w(x,y,t)$

at time t , the persistence value of that pixel would be higher than the value of another pixel which shows the presence of an object in infinite time instants in the past. Obviously, such a value assignment scheme is not desired. In another extreme case, d can get the value 1, which assigns equal significance to all the time instants including the current time instant and all the previous ones. Such a case is also not desirable as there should be some kind of priority for the recent time instants. In conclusion, the optimum values is somewhere between 0.5 and 1, where the significance of object presence decays gradually from the present to the past. Such an optimum value can be estimated by using a set of training videos as shown in section 9. Figure 3 shows the order of the permutations for the decay factor values of 0.5, 0.75 and 1. Observe how the order of permutations changes with d .

6. Global Persistence Threshold

As explained in previous sections, to distinguish real targets from noises in the scene, we rely on persistence of the objects on local neighborhoods throughout the sequence. The more persistent the objects are, the higher chance they have of being detected as real targets. Therefore a threshold is defined to determine when enough evidence is collected to approve an object as a real target. However, as the noise level of the scene increases, the chance of observing several noise objects in a small vicinity of the image increases. As a result, we need more evidence (i.e. longer history) of object presence to recognize an object as a target. This means a higher threshold. On the other hand, the target detection threshold is a function of the noise level of the scene. In the following, a method to estimate the global noise level of the scene as well as a way to determine the global persistence threshold is explained.

6.1. Global Noise Level Estimation

In many of the video sequences especially those of outdoor scenes, different regions might show different noise levels. However, most of this inconsistency is due to the presence of the noises of the second category as explained in section 3. Since this type of noise is the one we do not intend to deal with, we prefer to assign a global noise level to the entire region of the image, therefore we take the following steps to make a robust estimation:

1. Divide the image to a set of blocks of equal size.
2. At each block in the image, count the number of objects with the area smaller than a threshold $Area_{max}^{noise}$. This threshold is the maximum predicted area of each

noise object. An object of a larger area is accounted as a real target.

3. Compute the sum of object counts at each block over a certain time window T .
4. Find the median of all the sums, multiply it to the number of blocks in the image and divide it by the window size T .

The above algorithm gives us an estimate of the number of small objects (noise candidates) in the scene. Note that this number might include some small targets as well, but since the ratio of the number of targets to the number of noise objects is usually negligible, this would not have significant impact on our estimation. Figure 4 shows how the above steps work using some sample values.

6.2. Selecting the Global Persistence Threshold

The next step is turning the global noise level into a global persistence threshold which impacts all object-based thresholds to be determined later. In fact, all the real targets of any area, contrast or level of motion coherency need to show a longer persistency in a scene of high noise level than one with a lower noise level. The transformation of the scene noise level to a global threshold is through a non-linear function such as hyperbolic tangent or sigmoid. Figure 5 shows an example sigmoid function used for the experiments of this paper. As shown in the figure, if the noise level is smaller than a certain value, the global threshold would be 10...0 meaning that the object needs to be present just at the current frame to be detected as target. In contrast, 11...1 means that if the noise level is higher than a maximum value, it should have been present for at least T

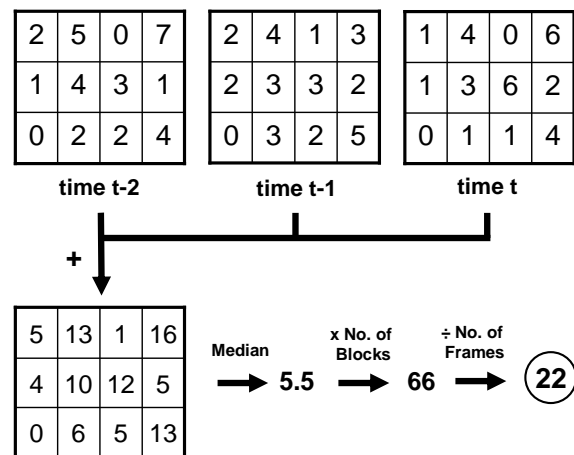
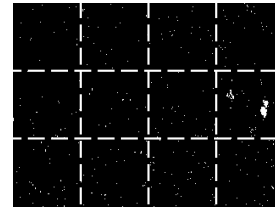


Figure 4. Noise Level Estimation: Top row: A sample image divided into 12 equal blocks. Middle row: Some sample values for the number of noise objects at each block for $T = 3$ consecutive frames. Bottom row: The noise level estimation process.

number of frames to be detected as target. For values between these two extremes, the sigmoid function determines the global threshold. It is worth noting that the values between 10...0 and 11...1 will not be in binary order but they follow a pattern defined by the decay factor d as discussed in section 5.1.

7. Object-based Analysis

Even though the global noise level is a major parameter in determining the persistence threshold for the targets, yet this threshold should vary from object to object. This is due to the fact that there are several parameters including the area and contrast of objects which can be relied on in the process of distinguishing target from noise. As a results, the object-based persistence threshold can change as a function

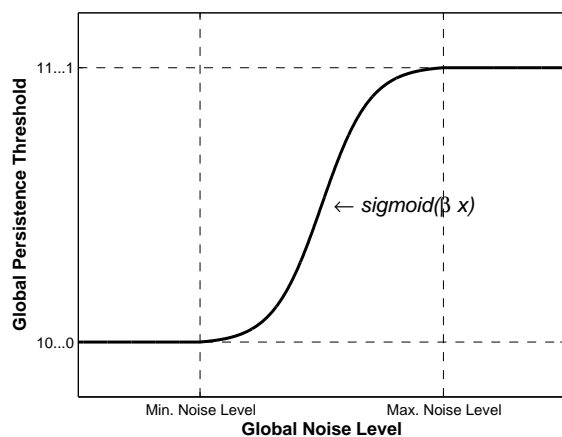


Figure 5. Converting global noise level to global persistence threshold.

of the area and contrast of each connected component. In this section we discuss first how to collect this information for each object and then how to select the threshold based on this information.

7.1. Object Property Measurement

Area and contrast of the objects are the two properties we need to measure in order to determine the object-based threshold. To do this measurement, we first create a set of binary foreground images by digitizing the output of a standard background subtraction method using different thresholds. Each one of these binary maps would include some portion of the area of real targets along with a set of noise objects. As shown in figure 1, the detected region of the real target shrinks as the threshold increases (compare the images of upper-right corner and lower-left corner). Meanwhile the number of noise objects would also reduce. This observation shows two things: First, the noise objects usually have low contrast which this complies with the common normal distribution usually used for noise modelling. Second, the region of the real targets in low-threshold maps are more complete and close to the reality. This is true even

considering that the region extracted from any of these maps is not necessarily a subset of the real target region as some noise objects might join the target region. In fact, due to the small size of noise objects, this expansion in size is negligible.

To measure the area and contrast of the objects in the scene, we first assign a number to each one of the foreground binary maps starting with 1 for the lowest threshold and then take the following steps for each object in the first map so called reference map:

1. Measure the area of the object by counting the 8-connected pixels making the object.
2. Find the map of the highest index with at least one pixel in common with the reference map. Assign the index of that map to the contrast of the object.

The process explained above creates some level of distinction between noise objects and real targets by assigning the two values of area and contrast. In fact, this approach detects a good portion of the target region by considering the map with the lowest threshold and assigns smaller values to the contrast of the noise objects which are likely to be missing in maps of higher thresholds. This distinction is improved by adding the persistency measure as explained in section 5.

It is worth noting that, instead of creating a set of binary foreground maps using discrete thresholds, we can use a single multi-valued map where the higher value shows higher contrast (i.e. larger foreground/background difference); however since in many of the background subtraction methods, the selection of a pixel as a foreground object is through combining the output of several parallel functions

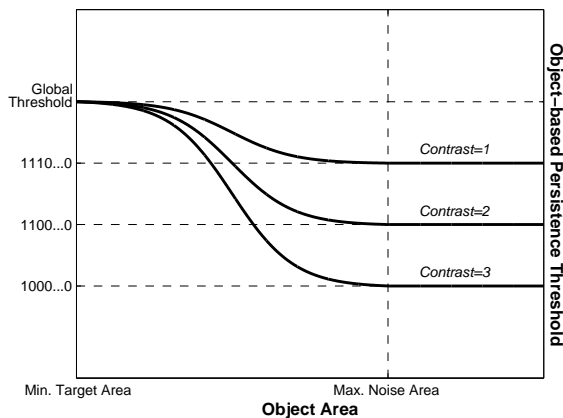


Figure 6. Object-base Threshold.

such as the mean and standard deviation of the difference, providing such a map might not be feasible.

7.2. Object-based Threshold Selection

Estimating the contrast and area of each object and also the global noise level of the scene, the persistence threshold for each object can be selected based on a non-linear function. Figure 6 shows a set of inverted sigmoid functions to be selected from based on the contrast of the object. As shown in the figure, the object-based persistence threshold is decreased by increasing the size of the object. However, the decrease factor depends on the contrast of the object. The values on the limit can change from the suggested values in the figure. Similar factor as β in the global threshold function need to be determined by experiments to adjust the speed of threshold increase from the highest value (*i.e.* global threshold) to the lowest ones.

8. Robustness of the Algorithm

The target detection algorithm presented in this paper shows a good level of robustness against different issues which usually cause problem for classical target detection

methods especially in noisy and low contrast videos. These issues and the proposed solutions to them follow:

1. Missing small and low-contrast objects: In night-time videos or in poor-illuminated regions in the scene, objects are usually low-contrast. Also, objects with significant distance to the camera are very small in size. In these cases, the appearance of the targets become close to the noise objects. As shown in this paper, persistence filter can still detect this type of targets as it does not rely simply on foreground/background difference thresholding.
2. Temporary Miss detections: In some frames in the scene due to the low-contrast, poor illumination, occlusion or particular color and texture of the background, some of the targets might not be detected in the preliminary background subtraction step. In such frames, the detection process fails; however our method is robust to these miss detections in a way that as soon as the targets reappear, it can recover their tracks. This is due to the fact that having some zeros in the vector $\mathbf{w}(x, y, t)$ is tolerable as long as $P(x, y, t)$ defined in equation 3 receives a higher value than the corresponding object-based threshold.
3. Sensitivity to the Parameters: There are several parameters to be determined in an object detection algorithm including the one proposed in this paper. Selecting suitable values for these parameters is a challenging task, because the values selected based on a set of training videos might not necessary work as well for a new video sequence captured at different illumination and noise level conditions. However, as it will be

shown in section 9, the persistence filter approach is not as sensitive to the value of its parameters as the similar approaches. The main reason for this insensitivity is that the algorithm is adaptive to the scene conditions and also the appearance and behavior of the objects.

In the next section, the effectiveness of the proposed method as well as the optimum values for the parameters are shown through several experiments on video sequences of different conditions.

9. Experimental Results

To measure the effectiveness of the proposed method, it was applied as an additional stage (filter) to a standard background subtraction method and the performance was compared between the cases when this module was disabled and the standard background subtraction acted (BS) and when the persistence filter module was enabled (PF). For this experiment, several night-time video sequences taken in different locations were used. For each sequence, the positions of the targets at each frame were manually selected¹ (i.e. ground truth) and compared with the target positions generated by the systems BS and PF. Then, the false alarms and miss detections were counted. The same experiment was performed using several different combinations of three major parameters of the background subtractor module including the distance thresholds for mean and standard deviation of the pixel intensities and also minimum area of the targets to detect. Figure 7 shows the normalized percentage of the false alarms and miss detections for different combi-

¹The ground truth data was generated using the Target Ground Truth Tool developed by ObjectVideo Inc.

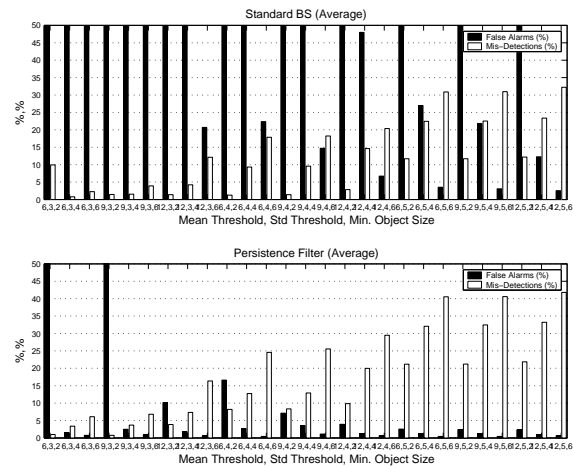


Figure 7. False alarm and miss detection Rate (Percent) for standard background subtraction algorithm and persistence filter averaged over a set of video clips.

nations of these three parameters averaged over all the sample videos. The normalized false alarms is computed as the ratio of the total number of false alarms to the total number of frames in the sequence. For miss detection, the ratio of the number of miss detections to the total number of target presences was used for normalization.

Comparing the results of the two approaches, the following observations can be made:

1. While there are some combinations of parameters where the BS method outperform the PF in either false alarm or miss detection rate, there is no case to be better in both. In fact, since lowering both the false alarm and miss detection rates matter in a vision system, persistence filter shows a significant improvement.
2. The persistence filter approach is more robust to the variation of the parameters. This makes it more applicable in different scenarios and illuminations.
3. Since the persistence filter has a filtering effect, for similar thresholds, the number of false alarms in PF

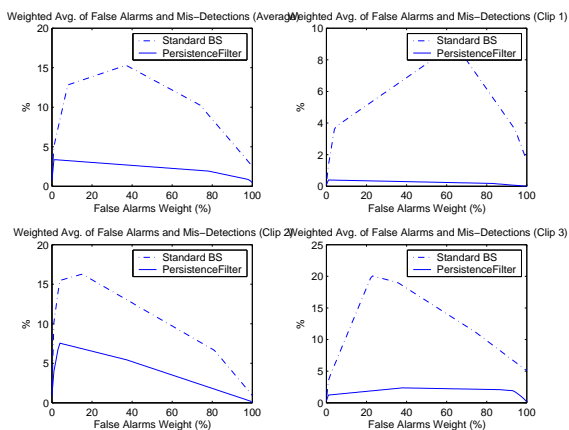


Figure 8. False alarm and miss detection Rate (Percent) for standard background subtraction algorithm and persistence filter averaged over a set of video clips.

is significantly smaller than BS whereas the number of miss detections is slightly larger. Therefore, using the persistence filter enabled system with lower thresholds provide us with a considerable performance improvement.

To be able to compare the two methods, a weighted average of the normalized false alarm and miss detection rates is defined as

$$f(FA, MD, \alpha) = \alpha FA + (1 - \alpha) MD \quad (7)$$

Figure 8 shows the function $f(\cdot)$ for both BS and PF methods for α value changing from 0 to 100. This shows no matter how important each one of the two objectives (i.e. lowering the false alarm rate of miss detection rate) is, the persistence filter outperform a classical background subtraction method.

10. Summary and Conclusions

A reliable method for detecting small targets in night-time videos was presented. Emphasizing on the inherent

differences of the noise and targets, we determine a detection threshold for each individual object while a global threshold calculated from the estimated noise level of the scene is used as an offset to these threshold. The Persistence map introduced in the paper represents the history of each object's presence. While the proposed method shows a comparable efficiency to the background subtraction methods, it provides a significant improvement in both false alarm and miss detection rate in night-time videos. By lowering the background subtraction thresholds and relying on the persistency of the objects, the method is also capable of detecting small and low contrast targets in the scene.

References

- [1] R. Cucchiara, C. Grana, M. Piccardi, and A. Prati. Detecting moving objects, ghosts, and shadows in video streams. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(10):1337–1342, 2003. 2
- [2] A. M. Elgammal, D. Harwood, and L. S. Davis. Non-parametric model for background subtraction. In *ECCV '00: Proceedings of the 6th European Conference on Computer Vision-Part II*, pages 751–767, London, UK, 2000. Springer-Verlag. 2
- [3] X. Gao, T. E. Boult, F. Coetzee, and V. Ramesh. Error analysis of background adaption. In *CVPR*, pages 1503–1510, 2000. 2
- [4] M. Heikkilä and M. Pietikäinen. A texture-based method for modeling the background and detecting moving objects. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(4):657–662, 2006. 2
- [5] O. Javed, K. Shafique, and M. Shah. A hierarchical approach to robust background subtraction using color and gradient information. In *MOTION '02: Proceedings of the Workshop*

1188		1242
1189	on <i>Motion and Video Computing</i> , page 22, Washington, DC,	1243
1190	USA, 2002. IEEE Computer Society. 2	1244
1191	[6] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. S. Davis.	1245
1192	Real-time foreground-background segmentation using code-	1246
1193	book model. <i>Real-Time Imaging</i> , 11(3):172–185, 2005. 2	1247
1194		1248
1195		1249
1196	[7] A. Mittal and N. Paragios. Motion-based background sub-	1250
1197	traction using adaptive kernel density estimation. In <i>CVPR</i>	1251
1198	(2), pages 302–309, 2004. 2	1252
1199		1253
1200		1254
1201	[8] T. Parag, A. Elgammal, and A. Mittal. A framework for	1255
1202	feature selection for background subtraction. <i>cvpr</i> , 2:1916–	1256
1203	1923, 2006. 2	1257
1204		1258
1205	[9] M. Piccardi. Background subtraction techniques: a review.	1259
1206	In <i>SMC (4)</i> , pages 3099–3104, 2004. 2	1260
1207		1261
1208		1262
1209	[10] J. Shao, S. K. Zhou, and R. Chellappa. Simultaneous back-	1263
1210	ground and foreground modeling for tracking in surveillance	1264
1211	video. In <i>ICIP</i> , pages 1053–1056, 2004. 2	1265
1212		1266
1213	[11] C. Stauffer and W. E. L. Grimson. Adaptive background mix-	1267
1214	ture models for real-time tracking. In <i>CVPR</i> , pages 2246–	1268
1215	2252, 1999. 2	1269
1216		1270
1217		1271
1218	[12] C. R. Wren, A. Azarbayejani, T. Darrell, and A. Pentland.	1272
1219	Pfinder: Real-time tracking of the human body. <i>IEEE Trans.</i>	1273
1220	<i>Pattern Anal. Mach. Intell.</i> , 19(7):780–785, 1997. 2	1274
1221		1275
1222		1276
1223		1277
1224		1278
1225		1279
1226		1280
1227		1281
1228		1282
1229		1283
1230		1284
1231		1285
1232		1286
1233		1287
1234		1288
1235		1289
1236		1290
1237		1291
1238		1292
1239		1293
1240		1294
1241		1295