

# Time Ordering Shuffling for Improving Background Subtraction

B. Laugraud, P. Latour, and M. Van Droogenbroeck

INTELSIG Laboratory, Department of Electrical Engineering and Computer Science,  
University of Liège, Belgium  
{b. laugraud, philippe. latour, m. vandroogenbroeck}@ulg.ac.be

**Abstract.** By construction, a video is a series of ordered frames, whose order is defined at the time of the acquisition process. Background subtraction methods then take this input video and produce a series of segmentation maps expressed in terms of foreground objects and scene background. To our knowledge, this natural ordering of frames has never been questioned or challenged.

In this paper, we propose to challenge, in a prospective view, the natural ordering of video frames in the context of background subtraction, and examine alternative time orderings. The idea consists in changing the order before background subtraction is applied, by means of shuffling strategies, and re-ordering the segmentation maps afterwards. For this purpose, we propose several shuffling strategies and show that, for some background subtraction methods, results are preserved or even improved. The practical advantage of time shuffling is that it can be applied to any existing background subtraction seamlessly.

**Keywords:** Background subtraction, motion analysis, change detection, time ordering shuffling, updating strategy.

## 1 Introduction

Background subtraction is an essential task in many video applications (such as video-surveillance), that consists in discriminating between moving objects and the background of the scene. While this operation seems obvious, it is very difficult to accommodate to the large variety of scenes and to deal with dynamical effects such as sudden lighting changes, dynamic backgrounds, shadows, etc. Therefore, many techniques have been proposed and reviewed for background subtraction (see for example Brutzer *et al.* [1], Bouwmans [2,3], or Jodoin *et al.* [4]).

Background subtraction involves a background model and several steps: initialization, segmentation, and updating. During the initialization, the algorithm builds a background model. We are not focusing on initialization in this paper. Therefore, we assume that the learning sequence is long enough to build a model of the background. After initialization, we perform a segmentation, that produces the binary foreground/background segmentation map, and update the

background model. The quality of the final segmentation maps is usually expressed in terms of False (F) or True (T) Negatives (N) or Positives (P), or with global measures such as the  $F_1$  score [5], defined as:

$$F_1 = \frac{TP}{TP + \frac{FN + FP}{2}}. \quad (1)$$

It is assumed that only an appropriate background model, which is dynamic by essence, can lead to a good performance. This implies an efficient adaptation of the model over time, which is the purpose of the updating step.

Updating the background model is challenging because, among others, there is a trade-off between the adaptability to fast changes and the required inertia to avoid the rapid inclusion of abandoned objects into the background. Consequently, the updating mechanism, which is intrinsically related to the segmentation step, is an essential part of any background subtraction process.

The goal of this paper is to question the impact of changing the time ordering of frames for the segmentation and updating phases; the flowchart of our method is given in Figure 1.

This paper is organized as follows. Section 2 describes several strategies for changing the natural time ordering. These strategies are evaluated on a selection of sequences that are part of the Change Detection 2014 dataset [6] (denoted CDnet 2014 hereafter), and results are reported in Section 3. Finally, Section 4 concludes this paper.

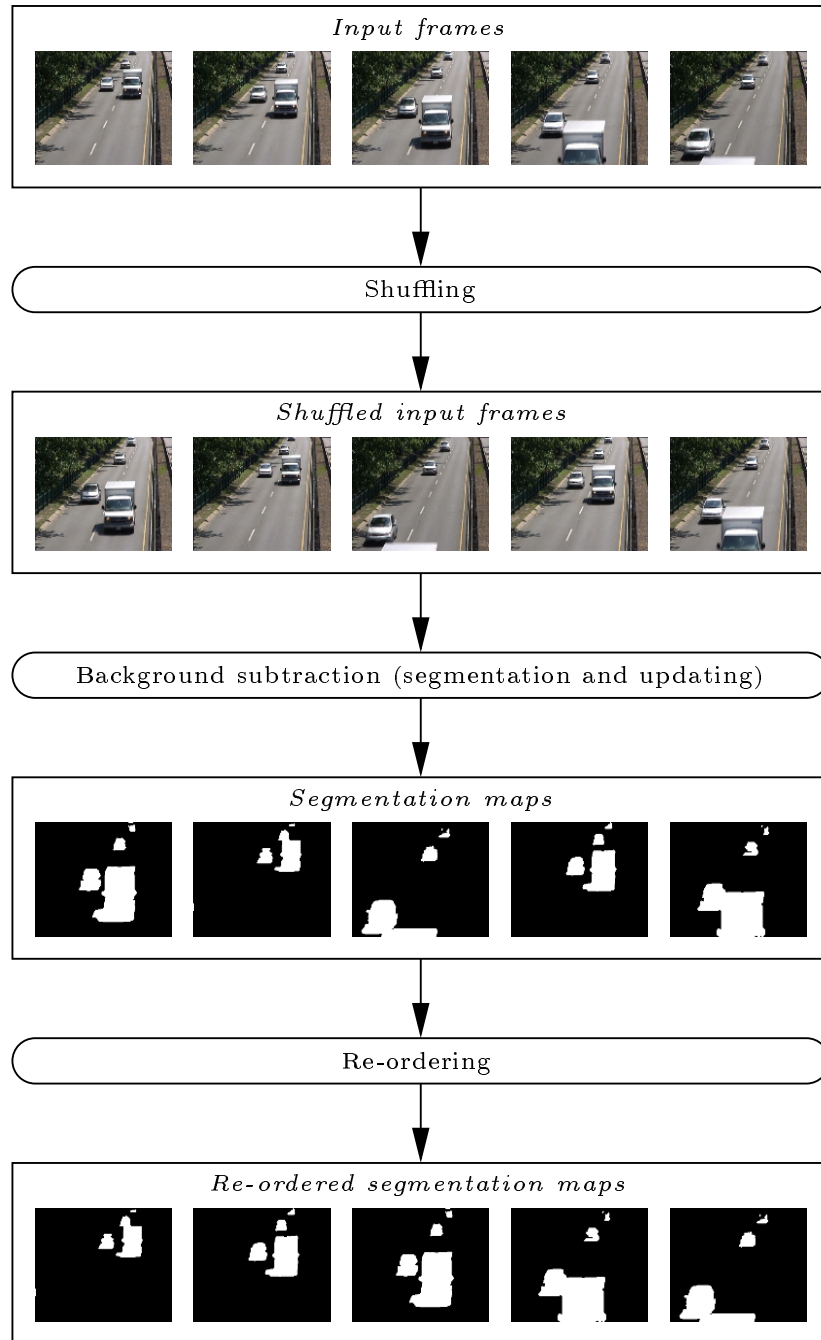
## 2 Time shuffling strategies

The natural order (which is the chronological one) is the most common order, although it might happen that some events are easier to interpret by running a video sequence backwards. In the field of background subtraction, some authors have recently tried to build background models that are less sensitive to the chronological order. For instance, while Wang and Suter have proposed a non parametric model that is updated by forgetting the oldest samples first [7], Barnich and Van Droogenbroeck rely on a random selection of samples to be replaced [8]; in this case, a video sequence is not seen as an *ordered list* of frames anymore, but rather as a *collection* of frames [9].

In the following, we first introduce the shuffling principle and then elaborate on a random policy and on deterministic shuffling strategies.

### 2.1 Principle

The time shuffling principle is the following. Let  $I$  be the ordered set of time indices of an  $N$ -frames long video sequence;  $I$  is then equal to  $\{1, 2, \dots, N\}$ , and the video  $V$  comprises the ordered frame sequence  $F_1 F_2 \dots F_N$ . The purpose of time shuffling consists to apply a *shuffling strategy* first, in order to obtain a new video whose frames are re-ordered. For example, after time shuffling,



**Fig. 1.** Flowchart of the proposed time shuffling scheme.

we have  $F_{13}F_2F_N \dots F_1$ . This new video sequence is used as an input for a background subtraction algorithm. This results in a video sequence made of the corresponding segmentation maps  $S_{13}S_2S_N \dots S_1$ . Finally, we recover the original order for the frames to obtain the output sequence  $S_1S_2 \dots S_N$ . The whole process is illustrated in Figure 1.

## 2.2 Random time shuffling strategy

In practice, we take for granted that the natural, chronological, time order is the most appropriate for dealing with videos. However, this has never been extensively checked. The opposite conception to the chronological time ordering is the absence of any order, that is, an entirely random ordering.

There are two reasons which justify the fact that we are interested in testing a random frame order. The first reason is that, since it can be seen as the opposite strategy to the forwards (chronological) ordering, we hope to span a large set of possible strategies and derive some insights on the importance of any kind of ordering. The second reason is that it is very difficult to imagine, at this stage of knowledge, a particular deterministic ordering. More precisely, if we introduce two of them in the next section, we remain unsure about the relevance of other non chronological deterministic strategies.

## 2.3 Deterministic time shuffling strategies

A more conservative approach consists in avoiding a random ordering. New strategies, that are deterministic, can be based on the content, on the time difference, or on both. In this paper, we examine two strategies based on the difference  $\delta$  between any pair of frames. Let  $F_1$  and  $F_2$  be two frames, we compute the difference as follows:

$$\delta(F_1, F_2) = \sum_{p \in h \times w} |F_1(p) - F_2(p)|, \quad (2)$$

where  $p$  is a pixel location taken in the frame of size  $h \times w$ .

Once we have computed all the differences, we choose an order that minimizes or maximizes the differences between successive images. The first strategy, named  $\delta_{\min}$ , is designed to minimize the amount of energy accumulated by  $\delta$ . The underlying justification for proposing this strategy is that some algorithms are designed to accommodate to small variations in the image content and that most algorithms have troubles for dealing with large changes.

The reason for the strategy that maximizes the differences (strategy named  $\delta_{\max}$  hereafter) is different. We want to evaluate if background subtraction techniques are performing better if they learn large changes as soon as possible, instead of learning them progressively. This would result in a model capable to embrace large changes earlier during the sequence, and a smooth adaptation to small changes later in the sequence.

Note that all the strategies are equally applicable to any type of background subtraction technique, because we do not change the techniques themselves.

Despite that, the choice of a strategy directly impacts on the performance of the updating scheme, as the methods are not all equally designed to deal with small or large changes.

## 2.4 Implementation of the deterministic strategies

Ideally, to implement our proposed deterministic strategies, we should look for the global optimum by finding the index order that minimizes or maximizes the cumulative difference energy. However, this exhaustive search is intractable. Thus, despite that, from a theoretical perspective, it is sub-optimal to proceed step by step (in other words, to find the new frame index at each step), an iterative algorithm has been implemented using this method to carry out the experiments of this paper.

Let  $\Omega_i$  and  $\Phi_i$  be two ordered sets of indices built after the  $i^{th}$  iteration (starting from 1), whose elements correspond to the remaining frames to shuffle and to the frames that have been shuffled, respectively. Considering the two initial sets  $\Omega_0 = I$  and  $\Phi_0 = \emptyset$ , the operations performed during the  $i^{th}$  iteration of the developed iterative algorithm are the following:

1. Select the element  $k_i \in \Omega_{i-1}$  which minimizes or maximizes the cumulative difference energy according to the chosen strategy. Note that the selection of  $k_1$  is discussed in Section 3.1.
2. Construct the set  $\Omega_i = \Omega_{i-1} \setminus \{k_i\}$ .
3. Construct the set  $\Phi_i = \Phi_{i-1} \cup \{k_i\}$ .

It should be noted that, applied on a  $N$ -frames long video sequence, the algorithm terminates after  $N$  iterations. In such a case, considering that  $P(I)$  is a permutation of  $I$ , we have that  $\Omega_N = \emptyset$ , and  $\Phi_N = P(I)$  which is the set of indices for the shuffled images.

In the example of Section 2.1, at the first iteration, once the frame indexed by 13 is selected, the sets  $\Omega_1 = \{1, \dots, 12, 14, \dots, N\}$  and  $\Phi_1 = \{13\}$  are built in such a way that this frame is removed from the set of the next candidates to select, and then included into the set of the shuffled frames.

## 3 Experiments and results

This section discusses preliminary results obtained for different shuffling strategies. First, we present how we deal with the initialization, then we discuss the chosen dataset, and finally we present our results.

### 3.1 Initialization

It is common acceptance that a background subtraction technique must initialize its model and that the first frames of a video sequence are used to build the model. Here, we have the choice to preserve the initial order for the initialization frames or to shuffle them. We decided to preserve the order during

the initialization step. By doing so, our experiments highlight the behavior of background subtraction techniques during the dynamic updating process only, without requiring any specific assumption about the initialization mechanism. As a consequence, we arbitrarily decided that the algorithm presented in Section 2.4 selects the first shuffled frame ( $k_1$ ) according to the differences  $\delta$  between the last initialization frame and the frames indexed by the elements of  $\Omega_0$ .

### 3.2 Dataset

In our experiments, we have used the CDnet 2014 dataset [6]. It is composed of 53 video sequences whose resolution varies from  $320 \times 240$  to  $720 \times 576$  pixels and duration from 600 to 7999 frames. The video sequences are provided with a ground truth annotated at the pixel level and they are divided among 11 categories to focus on specific challenges: Bad Weather, Baseline, Camera Jitter, Dynamic Background, Intermittent Object Motion, Low Framerate, Night Videos, PTZ, Shadow, Thermal, and Turbulence.

This dataset has several major advantages. First, as it is accompanied by ground truth segmentation maps, it is possible to perform an accurate evaluation [10] of our results. Second, since it contains many video sequences that cover a large variety of application scenarios, it offers the possibility to determine in which context the time ordering shuffling is relevant. And, last but not least, as it contains long sequences, it is more meaningful for the evaluation of time shuffling strategies than a dataset with very short sequences.

It should be noted that all the sequences of the CDnet 2014 dataset have been used, except for the ones from the PTZ category (where one model might fall short for dealing with the scenes), and the Low Framerate category (where motion between consecutive frames might be too large to understand the behavior of our shuffling strategies).

### 3.3 Background subtraction algorithms

In this paper, we consider six background subtraction algorithms, all of them operating at the pixel level, to challenge the relevance of time shuffling strategies. The simplest one, the exponential filter (denoted “Exp. Filter”), applies a temporal exponential filter on the observed value and thresholds the result. We have also used two mixture of Gaussians techniques, dedicated to the segmentation of scenes with dynamic backgrounds: the technique by Stauffer and Grimson [11] (“MoG G.”), and the one by Zivkovic, which adapts the number of Gaussians over time [12] (“MoG Z.”). We also consider the VuMeter algorithm, proposed by Goyat *et al.* [13], which is a non parametric method that estimates the background with a probability mass function. In this technique, the estimation is derived from the temporal histogram of the observed values.

The ViBe algorithm [8], proposed by Barnich and Van Droogenbroeck, is an alternative non parametric method which compares an observed value to a set of past samples using an Euclidean distance. Note that in order to update its model, ViBe combines a spatial propagation mechanism with a random sampling policy.

To our knowledge, ViBe is the first technique to consider a non chronological policy for updating the model. Therefore, it is also a good candidate for testing shuffling strategies.

The last method we have considered is an approach based on self organization through artificial neural networks, the so-called SOBS algorithm, as proposed by Maddalena and Petrosino [14], whose idea is inspired from biologically problem-solving methods. It should be noted that the implementations of these algorithms are provided by the BGSLibrary [15] or by the authors.

### 3.4 Results and interpretation

Obviously, shuffling frames if a scene cut or illumination change occurs is irrelevant because background models before and after such an event are different. Therefore, we only use video sequences that do not contain this type of events; sequences of the CDnet 2014 dataset comply with this assumption.

Likewise, one may wonder on the practical feasibility of a shuffling mechanism that would shuffle frames from the whole sequence. While we agree that, in practice, one should work with slices of the video sequence (by 100 frames for example), we perform experiments on the whole sequence because it helps to understand the impact of re-ordering frames prior to segmentation; by doing this on parts of the video sequence, we might not be able to evaluate the impact correctly. In addition, it should be noted that this problem of causality is nonexistent for off-line video processing.

**Table 1.**  $F_1$  scores for the forwards (*fw*) ordering and the random shuffling of frames (*rd*). While we believe that averaging  $F_1$  scores is questionable, we still provide the mean values for the whole dataset in the last row, for convenience.

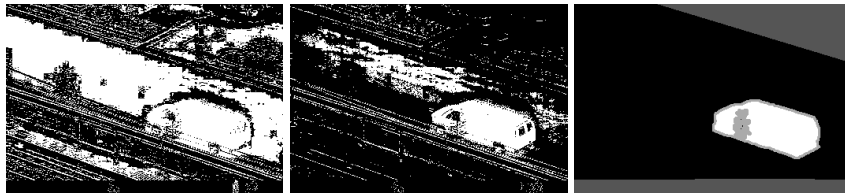
		Technique and strategy											
		Exp. Filter		MoG G.		MoG Z.		VuMeter		ViBe		SOBS	
		<i>fw</i>	<i>rd</i>	<i>fw</i>	<i>rd</i>	<i>fw</i>	<i>rd</i>	<i>fw</i>	<i>rd</i>	<i>fw</i>	<i>rd</i>	<i>fw</i>	<i>rd</i>
Category	Bad Weather	0.35	0.28	0.69	0.68	0.74	0.74	0.49	0.74	0.62	0.66	0.63	0.62
	Baseline	0.37	0.34	0.66	0.65	0.79	0.77	0.52	0.57	0.78	0.80	0.77	0.71
	Camera Jitter	0.23	0.22	0.45	0.47	0.51	0.51	0.49	0.52	0.54	0.54	0.44	0.44
	Dyn. Background	0.09	0.09	0.26	0.27	0.42	0.41	0.44	0.57	0.47	0.47	0.17	0.17
	Intermittent Obj.	0.29	0.31	0.43	0.49	0.46	0.49	0.20	0.31	0.41	0.46	0.51	0.51
	Night Videos	0.26	0.20	0.38	0.36	0.37	0.36	0.30	0.28	0.39	0.38	0.35	0.35
	Shadow	0.40	0.28	0.69	0.63	0.71	0.69	0.46	0.56	0.73	0.71	0.65	0.60
	Thermal	0.48	0.61	0.52	0.67	0.63	0.63	0.29	0.49	0.52	0.60	0.76	0.74
	Turbulence	0.07	0.07	0.30	0.25	0.53	0.52	0.54	0.53	0.67	0.57	0.17	0.17
	Dataset	0.28	0.27	0.49	0.50	0.57	0.57	0.41	0.51	0.57	0.58	0.50	0.48

In Table 1, we present the  $F_1$  scores for the set of background subtraction algorithms described in the Section 3.3. It is difficult to draw definitive conclusions from these experiments. In most cases, there are no big differences, despite that the random shuffling strategy is completely opposite to the chronological ordering. Only for the SOBS algorithm did we notice a significant performance decrease. We believe that this might be due to the small number of templates used in this technique and its inability to deal with a wide variety of background templates, such as that encountered when shuffling the video frames. Likewise, the exponential filter has a very limited memory capability. Therefore, it is counter-productive for it to try to estimate the model with shuffled frames.

In a second series of experiments, we compare the three new ordering strategies (random,  $\delta_{\min}$ , and  $\delta_{\max}$ ) and the chronological ordering (forwards) for a subset of categories only (it was impossible to check all our ideas, because there are too many possible deterministic strategies). Results for these experiments are given in Table 2 for the Baseline category (which gathers the most “standard” type of sequences), in Table 3 for the Camera Jitter category (which gathers sequences captured with unstable cameras), in Table 4 for the Dynamic Background category (which gathers outdoor sequences with moving backgrounds), and in Table 5 for the Intermittent Object Motion category (which gathers sequences with foreground objects that remains stationary for some time). With all the cautiousness required by our prospective experiments, we can formulate the following observations:

1. The  $\delta_{\max}$  strategy is usually not the best one. This means that algorithms for background subtraction are intrinsically mainly designed to handle small changes instead of large ones. This is not really surprising, but these experiments seem to confirm that intuition.
2. We observed that the  $\delta_{\min}$  strategy tends to mimic the forwards ordering, except when large changes occur in the image. Since computing  $\delta_{\min}$  is time consuming, it appears that the forwards ordering is more adequate. However, one should observe that  $\delta_{\min}$  always improves the segmentation of the sequences belonging to the Camera Jitter category (Table 3). This is coherent with the fact that background subtraction techniques are better in adapting to small changes.
3. The forwards strategy is never a clear winner for the video sequences of the Camera Jitter category (Table 3) and at least one of shuffling strategies improves the performance with respect to that of the natural ordering. This could be explained by the fact that shuffling the frames might reduce the amount of differences, due to camera motion, between consecutive frames (this is, again, coherent with our observation for the  $\delta_{\min}$  strategy). Figure 2 illustrates the improvement obtained by the use of a shuffling strategy on a sequence of the Camera Jitter category.
4. Except for the SOBS algorithm, the forwards strategy is never the best for the video sequences of the Intermittent Object Motion category (Table 5). Our feeling is that, because random and  $\delta_{\max}$  strategies will interlace frames with and without stationary objects, they slow down the inclusion of those

- objects into the background (which is considered as being the right approach by the designers of the CDnet dataset).
5. For most methods, the random strategy outperforms the natural ordering in the Thermal category (Table 1). Figure 3 illustrates the improvement obtained by the use of a shuffling strategy on a sequence of this category.
  6. For the VuMeter algorithm, in all cases, at least one shuffling strategy increases the score of the chronological ordering.
  7. For the Baseline category, there is no clear winning strategy. It is interesting to note that ViBe, which natively incorporates a random updating policy, performs best for the random shuffling strategy. This suggests that the sequences have backgrounds with a lot of inertia, so that the modification of the time order has no large impact on the performances.
  8. It is worth considering alternative time orderings as the forwards strategy is not always the best strategy. This raises questions about the dynamic behavior of background subtraction techniques in general.



**Fig. 2.** Segmentation of frame numbered 820 of the *boulevard* sequence (Camera Jitter category) by the MoG G. algorithm using the forwards (left-hand side image) and  $\delta_{\max}$  (middle image) shuffling strategies. The last column shows the ground truth.



**Fig. 3.** Segmentation of frame numbered 2018 of the *corridor* sequence (Thermal category) by the VuMeter algorithm using the forwards (left-hand side image) and random (middle image) shuffling strategies. The last column shows the ground truth.

**Table 2.**  $F_1$  scores for different shuffling strategies applied on the Baseline category.

		Strategy			
		forwards	random	$\delta_{\min}$	$\delta_{\max}$
Technique	Exp. Filter	<b>0.374</b>	0.342	<b>0.374</b>	0.349
	MoG G.	<b>0.658</b>	0.657	<b>0.658</b>	0.653
	MoG Z.	<b>0.794</b>	0.778	0.791	0.782
	VuMeter	0.523	<b>0.572</b>	0.525	0.489
	ViBe	0.777	<b>0.801</b>	0.778	0.790
	SOBS	<b>0.771</b>	0.710	0.760	0.755

**Table 3.**  $F_1$  scores for different shuffling strategies applied on the Camera Jitter category.

		Strategy			
		forwards	random	$\delta_{\min}$	$\delta_{\max}$
Technique	Exp. Filter	0.228	0.222	<b>0.265</b>	0.237
	MoG G.	0.452	0.469	0.504	<b>0.505</b>
	MoG Z.	0.505	0.505	<b>0.515</b>	0.498
	VuMeter	0.486	<b>0.519</b>	0.503	0.498
	ViBe	0.542	0.539	<b>0.558</b>	0.537
	SOBS	0.437	0.437	0.439	<b>0.443</b>

**Table 4.**  $F_1$  scores for different shuffling strategies applied on the Dynamic Background category.

		Strategy			
		forwards	random	$\delta_{\min}$	$\delta_{\max}$
Technique	Exp. Filter	<b>0.094</b>	0.091	<b>0.094</b>	0.090
	MoG G.	0.258	<b>0.270</b>	0.258	0.260
	MoG Z.	<b>0.420</b>	0.410	<b>0.420</b>	0.410
	VuMeter	0.441	<b>0.572</b>	0.441	0.417
	ViBe	0.472	<b>0.473</b>	0.471	0.463
	SOBS	<b>0.169</b>	0.166	<b>0.169</b>	0.167

**Table 5.**  $F_1$  scores for different shuffling strategies applied on the Intermittent Object Motion category.

		Strategy			
		forwards	random	$\delta_{\min}$	$\delta_{\max}$
Technique	Exp. Filter	0.287	<b>0.306</b>	0.284	0.302
	MoG G.	0.426	<b>0.483</b>	0.414	0.454
	MoG Z.	0.461	<b>0.490</b>	0.462	0.461
	VuMeter	0.197	<b>0.309</b>	0.200	0.204
	ViBe	0.408	<b>0.460</b>	0.408	0.431
	SOBS	<b>0.514</b>	0.508	<b>0.514</b>	0.508

## 4 Conclusions

In this prospective paper, we examine how changing the time ordering of video frames impacts on the the task of background subtraction. This questions the notion of chronological order and introduces a new view on the problem. For that purpose, we have developed several shuffling strategies for re-ordering the frames and have evaluated them for different algorithms on the CDnet 2014 dataset. This highlights whether techniques are or not dependent with respect to the frame order, but it also shows that a principle applicable to any background subtraction technique can affect the performance. Our results remain difficult to interpret, because they illustrate that the updating process of some algorithms is sensitive to the time ordering of the frames, while other techniques are less sensitive to it. In addition, we observe that the chronological ordering is rarely the best one for background subtraction, and that for some categories of video sequences, at least one shuffling strategy systematically improves the  $F_1$  score. While this is a question to be challenged in future work, it also points out that the general understanding of background subtraction updating mechanisms has to be refined.

## References

1. Brutzer, S., Höferlin, B., Heidemann, G.: Evaluation of background subtraction techniques for video surveillance. In: IEEE International Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1937–1944 (2011), <http://dx.doi.org/10.1109/CVPR.2011.5995508>
2. Bouwmans, T.: Recent advanced statistical background modeling for foreground detection - A systematic survey. *Recent Patents on Computer Science* 4(3), 147–176 (2011), <http://dx.doi.org/10.2174/2213275911104030147>
3. Bouwmans, T.: Traditional and recent approaches in background modeling for foreground detection: An overview. *Computer Science Review* 11-12, 31–66 (2014), <http://dx.doi.org/10.1016/j.cosrev.2014.04.001>
4. Jodoin, P.-M., Piérard, S., Wang, Y., Van Droogenbroeck, M.: Overview and benchmarking of motion detection methods. In: Bouwmans, T., Porikli, F., Hoferlin, B., Vacavant, A. (eds.) *Background Modeling and Foreground Detection for Video Surveillance*, chap. 24. Chapman and Hall/CRC (2014), <http://www.crcnetbase.com/doi/abs/10.1201/b17223-30>

5. Baumann, A., Boltz, M., Ebling, J., Koenig, M., Loos, H., Merkel, M., Niem, W., Warzelhan, J., Yu, J.: A review and comparison of measures for automatic video surveillance systems. *EURASIP Journal on Image and Video Processing* p. 30 pages (2008), <http://dx.doi.org/10.1155/2008/824726>
6. Wang, Y., Jodoin, P.-M., Porikli, F., Konrad, J., Benezeth, Y., Ishwar, P.: CDnet 2014: An expanded change detection benchmark dataset. In: *IEEE International Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. pp. 393–400 (2014), <http://dx.doi.org/10.1109/CVPRW.2014.126>
7. Wang, H., Suter, D.: A consensus-based method for tracking: Modelling background scenario and foreground appearance. *Pattern Recognition* 40(3), 1091–1105 (2007), <http://dx.doi.org/10.1016/j.patcog.2006.05.024>
8. Barnich, O., Van Droogenbroeck, M.: ViBe: A universal background subtraction algorithm for video sequences. *IEEE Transactions on Image Processing* 20(6), 1709–1724 (2011), <http://dx.doi.org/10.1109/TIP.2010.2101613>
9. Van Droogenbroeck, M., Barnich, O.: Visual background extractor. World Intellectual Property Organization, WO 2009/007198, 36 pages (2009), <https://patentscope.wipo.int/search/en/detail.jsf?docId=WO2009007198>
10. Goyette, N., Jodoin, P.-M., Porikli, F., Konrad, J., Ishwar, P.: A novel video dataset for change detection benchmarking. *IEEE Transactions on Image Processing* 23(11), 4663–4679 (2014), <http://dx.doi.org/10.1109/TIP.2014.2346013>
11. Stauffer, C., Grimson, E.: Adaptive background mixture models for real-time tracking. In: *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*. vol. 2, pp. 246–252 (1999), <http://dx.doi.org/10.1109/CVPR.1999.784637>
12. Zivkovic, Z.: Improved adaptive Gaussian mixture model for background subtraction. In: *IEEE International Conference on Pattern Recognition (ICPR)*. vol. 2, pp. 28–31 (2004), <http://dx.doi.org/10.1109/ICPR.2004.1333992>
13. Goyat, Y., Chateau, T., Malaterre, L., Trassoudaine, L.: Vehicle trajectories evaluation by static video sensors. In: *IEEE Intelligent Transportation Systems Conference (ITSC)*. pp. 864–869 (2006), <http://dx.doi.org/10.1109/ITSC.2006.1706852>
14. Maddalena, L., Petrosino, A.: A self-organizing approach to background subtraction for visual surveillance applications. *IEEE Transactions on Image Processing* 17(7), 1168–1177 (2008), <http://dx.doi.org/10.1109/TIP.2008.924285>
15. Sobral, A.: BGSLibrary: An OpenCV C++ background subtraction library. In: *Workshop de Visao Computacional (WVC)* (2013), <http://dx.doi.org/10.13140/2.1.1740.7044>