Change Detection for Temporal Texture in the Fourier Domain

Alexia Briassouli and Ioannis Kompatsiaris

Informatics and Telematics Institute
Centre for Research and Technology, Hellas
6th km, Charilaou-Thermis
Thermi, 57001

Abstract. Research on temporal textures has concerned mainly modeling, synthesis and detection, but not finding changes between different temporal textures. Shot change detection, based on appearance, has received much research attention, but detection of changes between temporal textures has not been addressed sufficiently. Successive temporal textures in a video often have a similar appearance but different motion, a change that shot change detection cannot discern. In this paper, changes between temporal textures are captured by deriving a non-parametric statistical model for the motions via a novel approach, based on properties of the Fourier transform. Motion statistics are used in a sequential change detection test to find changes in the motion distributions, and consequently the temporal textures. Experiments use a wide range of videos of temporal textures, groups of people, traffic. The proposed approach leads to correct change detection, at a low computational cost.

1 Introduction

The analysis of motion in video is fundamental for characterizing its contents, motion segmentation, activity recognition, and various other tasks. Temporal textures, also known as dynamic textures, are videos of textures which evolve over time. Numerous methods have been developed for the modeling and subsequent analysis of temporal textures, based on appropriately designed features of the video, or a model of the temporal texture process [1]. Initial work on temporal textures modeled them based on a combination of spatial and flow features [2]. Model based techniques have been developed for the recognition and synthesis of dynamic textures. The Spatiotemporal Auto-Regressive model (STAR) has been used extensively [3], while improvements have been made on it to account for rotational, non-translational motions [4]. In [5], temporal texture motions are modeled as a random process by using adaptive, predictive models. In [6], they are modeled by a linear dynamical system, and then spatially segmented by generalized PCA. A similar problem, that of spatially segmenting multiple temporal textures in each video frame, is addressed in [7], where motion co-occurrence statistics are used, as they provide a correspondence between the features of single and multiple textures.

Proposed approach: This work presents a novel method for the detection of changes in videos of successive temporal textures. We refer to each sequence of frames that corresponds to a temporal texture as a "temporal texture subsequence". The videos considered contain one temporal texture over a series of video frames, followed by a different temporal texture over another series of frames. The proposed method detects the frame at which a change occurs from one temporal texture subsequence to the next. For example, the frame at which the rippling of leaves changes intensity or density, the frame at which a crowd's motion changes, are found. These changes are difficult to detect by traditional shot change detection methods, which are most often based on appearance, because in many practical applications, such as detecting the change in the flow of traffic in a highway, there is a very small differentiation in the appearance of successive temporal textures. Traditional motion estimation methods are not suited for such videos either, as the motions of temporal textures are highly non-rigid [8]. This paper addresses the problem of separating temporal textures based on their (different) motion characteristics, rather than appearance features. The motion is modeled as following a random distribution based on properties of the FT, as detailed in Sec. 2, 3. The resulting statistical model is non-parametric and offers a complete description of the motion characteristics of each temporal texture. It is used to detect when a change takes place in the video from one temporal texture to another via sequential change detection techniques, which are designed precisely for the problem of detecting a change from one distribution to the next, by using each sample (video frame) as it arrives.

Comparison with existing methods: In [5], change detection refers to the separation of a foreground undergoing rigid object motion from non-rigid background motion via an adaptive predictive model for the temporal textures. Temporal texture motion field estimation takes place in [9] based on the STAR model, giving an accurate flow field, but at a high computational cost. In this work, the motion field is approximated as a random vector following a non-parametric probability distribution that is extracted at a low cost from the Fourier Transform (FT) of the video (Sec. 2, 3). This provides sufficient information for the separation of different temporal texture subsequences, without the computational burden of estimating a precise motion field. In [10], the change point between successive temporal textures is detected by modeling the video by a linear dynamic system whose parameters are determined after training with the EM algorithm. Our work differs from that of [10], as we derive a non-parametric description of the temporal texture motion, derived empirically from the data. Thus, we avoid issues such as tuning the EM algorithm parameters to ensure convergence, or determining the optimal clustering technique for segmenting the video into similar temporal textures, that is required in [10], [11]. The non-parametric model derived in this work provides a better description of the motion statistics than a simple histogram of the motion vectors, as the latter are not easy to calculate with accuracy and speed in temporal texture videos. Both approaches present a solution to the challenging problem of separating temporal textures, which cannot be dealt with the conventional optical flow or appearance based video segmentation methods, due to the stochasticity of the motion in temporal textures, its high non-rigidity and the large number of moving entities present in them (e.g. tree leaves, people in a crowd etc.).

This paper is organized as follows. In Sec. 2, a statistical model for a video containing multiple random motions or, equivalently, temporal textures, is presented. Sec. 3 provides the theoretical description of the method proposed for approximating the random motion distribution, based on the video FT. Sequential change detection for finding the moments of change between temporal texture subsequences is presented in Sec. 4. Experimental results for videos with various temporal texture subsequences, crowds of people walking, highway traffic, are shown in Sec. 5, and conclusions are provided in Sec. 6.

2 Statistical Model for Multiple Random Motions, Temporal Textures

We consider the case of a video containing successive temporal texture subsequences. Each subsequence contains a single temporal texture over all frame pixels, i.e. we do not examine the case where there are more than one temporal textures present in a frame (as in [7]). The pixel intensity of frame k of the video sequence at pixel $\bar{r}$ is represented by $a(\bar{r}, k)$, and is considered to consist of M moving "objects" $s_i(\bar{r}),\ 1 \le i \le M$. The areas of the video that do not contain a moving object, e.g. the sky behind fluttering tree leaves, belong to background pixels, whose illumination is denoted as $s_b(\bar{r})$. Then, frame 1 is given by:

$$a(\bar{r}, 1) = s_b(\bar{r}) + s_1(\bar{r}) + ... + s_M(\bar{r}). \tag{1}$$

In videos of a group of people or of traffic, the moving objects $s_i(\bar{r})$ are the people or cars. In videos of temporal textures like flowing water or leaves moving in the wind, the M objects represent each moving pixel or small group of pixels (e.g. the pixels forming a leaf), since those motions are highly non-rigid. In reality, the relation between the video frame illumination and that of the background and the objects' illumination is not additive, since the object pixels actually mask background pixels. However, in many temporal texture videos, there is no background at all (all pixels are moving e.g. in a video of flowing water). Even in cases of temporal textures with a background, the random movements are small, so the resulting occlusion is insignificant. This is verified by the experimental results with various types of backgrounds, where the change detection provides accurate results. The Fourier transform (FT) of the first video frame is given by:

$$A(\bar{\omega}, 1) = S_b(\bar{\omega}) + S_1(\bar{\omega}) + ... + S_M(\bar{\omega}), \tag{2}$$

where $\bar{\omega}$ is the two-dimensional frequency of the FT. At frame k, every object (group of pixels corresponding to a group of people, cars, or a single leaf, flower on a tree) has been displaced by random $\bar{r}_i,\ 1 \le i \le M$, so frame k is given by:

$$a(\bar{r}, k) = s_b(\bar{r}) + s_1(\bar{r}, k) + ... + s_M(\bar{r}, k) = s_b(\bar{r}) + s_1(\bar{r} - \bar{r}_1) + ... + s_M(\bar{r} - \bar{r}_M), \tag{3}$$

and since $S_i(\bar{\omega}, k) = S_i(\bar{\omega})e^{-j\bar{\omega}^T \bar{r}_i}$, its FT becomes:

$$A(\bar{\omega}, 1) = S_b(\bar{\omega}) + S_1(\bar{\omega})e^{-j\bar{\omega}^T \bar{r}_1} + ... + S_M(\bar{\omega})e^{-j\bar{\omega}^T \bar{r}_M}. \tag{4}$$

It is known from probability theory that the characteristic function of a probability density function is equal to the complex conjugate of its FT [12], i.e.

$$\Phi(\bar{\omega}) = \mathfrak{F}^*[f(\bar{r})] = \int_{-\infty}^{+\infty} f(\bar{r})e^{j\bar{\omega}^T \bar{r}} d\bar{r} = E[e^{j\bar{\omega}^T \bar{r}}]. \tag{5}$$

For videos of temporal textures, each displacement $\bar{r}_i$ is considered to follow a random distribution $f_i(\bar{r})$.The expected value of eq. (4) is then given by:

$$\begin{aligned} E[A(\bar{\omega}, k)] &= E[S_b(\bar{\omega})] + E[S_1(\bar{\omega})e^{-j\bar{\omega}^T \bar{r}_1}] + ... + E[S_M(\bar{\omega})e^{-j\bar{\omega}^T \bar{r}_M}] \\ &= S_b(\bar{\omega}) + S_1(\bar{\omega})E[e^{-j\bar{\omega}^T \bar{r}_1}] + ... + S_M(\bar{\omega})E[e^{-j\bar{\omega}^T \bar{r}_M}] \\ &= S_b(\bar{\omega}) + S_1(\bar{\omega})\Phi_1^*(\bar{\omega}) + ... + S_M(\bar{\omega})\Phi_M^*(\bar{\omega}), \end{aligned} \tag{6}$$

where the expected value operator $E[\cdot]$ only affects the displacements since the other quantities are not random, and where we have used the definition of eq. (5) to include the characteristic function in eq. (6).

The characteristic function provides the most complete statistical description of a random variable, as that random variable's pdf and all its existing moments can be derived from the characteristic function. The probability density function $f_i(\bar{r})$ corresponding to each characteristic function $\Phi_i(\bar{\omega})$ is given by:

$$f_i(\bar{r}) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \Phi_i(\bar{\omega})e^{-j\bar{\omega}^T \bar{r}} d\bar{\omega}. \tag{7}$$

Thus, the expression of each video frame's FT as a function of its random motion's characteristic function provides a comprehensive description of the activity in it, motivating us to use it for detecting changes between temporal texture subsequences.

3 Statistical Model for One Type of Random Motion, Temporal Textures

In this work we focus on videos containing one kind of random motion, or one kind of temporal texture, in the frames of a temporal texture subsequence. This encompasses a wide range of temporal texture videos found in practical applications, such as videos of trees, water flowing, groups of people walking, and traffic. Since one kind of random motion takes place, we have $\bar{r}_i \sim f_0(\bar{r})$ with characteristic function $\Phi_0(\bar{\omega})$, and the model of eq. (6) can be re-written as:

$$E[A(\bar{\omega}, k)] = S_b(\bar{\omega}) + S_1(\bar{\omega})\Phi_0^*(\bar{\omega}) + ... + S_M(\bar{\omega})\Phi_0^*(\bar{\omega}) = S_b(\bar{\omega}) + \Phi_0^*(\bar{\omega}) \sum_{i=1}^{M} S_i(\bar{\omega}). \tag{8}$$

For videos with a static background or a background that undergoes small changes, $S_b(\bar{\omega})$ can be removed via one of the numerous background removal techniques that are available [13], [14]. In practice, very many videos of temporal textures contain no background, or a very small static background area, so its removal is either very simple or not necessary at all. Consequently, the background can be removed or ignored (if it is very small or non-existent) and eq. (8) can be written as:

$$E[A(\bar{\omega}, k)] = \Phi_0^*(\bar{\omega}) \sum_{i=1}^{M} S_i(\bar{\omega}). \tag{9}$$

The FT of the first frame of such a video is given by:

$$E[A(\bar{\omega}, 1)] = \sum_{i=1}^{M} S_i(\bar{\omega}). \tag{10}$$

By combining eqs. (9) and (10), we have:

$$\frac{E[A(\bar{\omega}, k)]}{E[A(\bar{\omega}, 1)]} = \frac{\Phi_0^*(\bar{\omega}) \sum_{i=1}^{M} S_i(\bar{\omega})}{\sum_{i=1}^{M} S_i(\bar{\omega})} = \Phi_0^*(\bar{\omega}). \tag{11}$$

Thus, if the expected values of the frames' FTs $E[A(\bar{\omega}, 1)]$, $E[A(\bar{\omega}, k)]$ are known, we can obtain a complete description of the random motion in the temporal texture video. The video frames are available and their FT in Eq. (11) can easily be computed. In order to estimate the expected value of these FTs, several instantiations of each temporal texture video need to be available under the ergodicity assumption. However, this is not practically feasible, as usually only one instance of each video is available. In order to overcome this issue, we make the observation that neighboring video frames are characterized by similar motions, so a subsequence of w_0 frames can be considered to consist of instantiations of a random motion that follows the distribution f_0. Thus, the expected value of $A(\bar{\omega}, 1)$ and $A(\bar{\omega}, k)$ can be approximated as follows:

$$E[A(\bar{\omega}, 1)] = \frac{1}{w_0} \sum_{i=1}^{w_0} A(\bar{\omega}, i), \quad E[A(\bar{\omega}, k)] = \frac{1}{w_0} \sum_{i=k}^{k+w_0-1} A(\bar{\omega}, i), \tag{12}$$

leading to the approximation of the motion characteristic function near frame k:

$$\Phi_0(\bar{\omega}) = \frac{\sum_{i=k}^{k+w_0-1} A^*(\bar{\omega}, i)}{\sum_{i=1}^{w_0} A^*(\bar{\omega}, i)}. \tag{13}$$

As mentioned in Sec. 2, and known from probability theory [12], the probability density function can be obtained from the characteristic function, so by estimating $\Phi_0(\bar{\omega})$, we also know the pdf $f_0(\bar{r})$. The change detection algorithm presented in the section that follows is based on knowledge of this pdf. The pdf of Eq. (13) is extracted from the ratio of FTs, so only the information in the FT phase is retained: all appearance information is eliminated, and the proposed method uses only on the extracted *motion* distribution information.

4 Change Detection for Temporal Textures

As mentioned in Sec. 1, this work focuses on the detection of the moments of change in videos of temporal textures. Its aim is to detect at which frame the random motion changes, i.e. the motion follows a different pdf. For example, the change between water flowing slowly and then fast, traffic changing from heavy to light, is to be detected. It should be noted that the problem addressed here is particularly challenging, as not only the moment of change M is unknown, but the motion distributions before and after the change are also not known. Sequential change detection methods are well suited to this task, since they can detect changes in a sequence of random variables as they arrive. For the data sequence $\bar{x} = [x_1, x_2, ..., x_N]$, where the first L variables follow a pdf f_0, while the rest follow pdf f_1, the following hypothesis test applies:

$$\begin{aligned} H_0 &: x_n \sim f_0 \\ H_1 &: x_n \sim f_1. \end{aligned} \tag{14}$$

Sequential change detection examines the data for a change by estimating the log-likelihood ratio (LLRT) of the input data, as proposed by Page in [15], in order to detect if a change has happened at each frame k. For the data from frames 1 to k, the LLRT is given by:

$$T_{1,k} = \ln \frac{f_1(\bar{x}_{1,k})}{f_0(\bar{x}_{1,k})}, \tag{15}$$

where $\bar{x}_{1,k} = [x_1, ..., x_k]$ are all the samples from frames 1 to k. If these samples are independent and identically distributed (i.i.d.), $T_{1,k}$ becomes:

$$T_{1,k} = \ln \prod_{n=1}^{k} \frac{f_1(x_n)}{f_0(x_n)} = \sum_{n=1}^{k} \ln \frac{f_1(x_n)}{f_0(x_n)}. \tag{16}$$

The i.i.d. assumption is reasonable when neighboring pixels in a video frame move independently from each other, as is the case in temporal textures videos, which usually contain highly non-rigid activity. Additionally, the i.i.d. assumption is often necessary in practice, as the joint pdf of the motion over all data samples can be very cumbersome and impractical to estimate. For the case of i.i.d. samples, the test statistic of Eq. (16) is expressed in a computationally efficient iterative form [15] as a cumulative sum, giving the CUSUM test:

$$T_k = \max\left(0, T_{k-1} + \ln \frac{f_1(x_n)}{f_0(x_n)}\right), \tag{17}$$

where $T_k = T_{1,k}$ and $T_0 = 0$. This test detects a change when the distribution changes from f_0 to f_1, making the test statistic T_k higher than a pre-defined threshold. Currently, there is no generally applicable way to obtain a closed-form expression for the threshold that will lead to the highest change detection rates, with the smallest amount of false alarms. A method for learning the threshold sequentially has been presented in [16], where the threshold is the solution to a diffusion equation. However, that approach is limited to computing a threshold for only a few simple cases, and also requires training with samples that follow both H_0 and H_1, which is often impractical. In [17], another method of threshold learning for the CUSUM test is proposed, based on parametric models for $f_1 = f(\theta_1), f_0 = f(\theta_0)$, but requires a non-parametric empirical approximation of the probability distributions before and after a change. Thus, the thresholds for the most general case are determined empirically, by experimental tuning with training data [18]. Here, after training with several videos, it is found that the threshold at frame k can be obtained by the formula:

$$\eta = \mu_k + c \cdot \sigma_k, \tag{18}$$

where μ_k, σ_k are the mean and standard deviation, respectively, of the test statistic from frames 1 to $k-1$. When there is a change, T_k becomes significantly higher than its previous values, and consequently the threshold η, which takes them into account. Experiments with this formula applied to videos in the categories included in our experiments show that correct detection but few false alarms are obtained for $c = 2$ to $c = 3$.

In order to implement the test of eq. (17), the pdfs f_0 and f_1 need to be known. They can be estimated from their characteristic functions that are derived as described in Section 3. In order to calculate f_0, or equivalently Φ_0, we make the assumption that the first w_0 frames of the video under examination correspond to the initial motion, with "baseline" distribution f_0. In order to approximate f_1, we assume that frame k and its neighboring w_0 frames follow f_1, and use this data in eq. (13). As mentioned in Sec. 3, it is reasonable to assume that in neighboring frames similar motion is taking place. This approach is common in similar problems [19], where the online estimation of distributions is necessary because of the complete lack of knowledge regarding them. When a change occurs, this assumption is violated temporarily, until all data being used follows the new distribution.

If the data from frames $k - w_0 + 1$ to k actually follows f_0, it will produce a "current" distribution approximation f_1 which is close to f_0, leading to a low value of the LLRT in eq. (17). When a change has occurred, the pdf approximation for f_1 will deviate from that for f_0, leading to a higher value for the LLRT, which will indicate that a change took place. It should be noted that the CUSUM test has been proven to provide the fastest detection of change, so there will not be a significant delay between the changepoint and the instant at which the test statistic T_k becomes higher than the detection threshold.

5 Experiments

Many different kinds of video sequences are examined to test the proposed system, and compare it with shot change detection. In the experiments the shot change detection fails to detect changes, whereas our approach can find meaningful changes. Table 1 contains the detected changes for the proposed approach and shot change detection. The proposed method runs faster than shot change detection, taking a fourth of the time to run, although it is implemented in Matlab, while the shot detection is implemented in C++. This can be attributed to the simpler nature of the algorithm.

5.1 Temporal Textures

A series of videos containing subsequences of temporal textures that undergo changes is examined. These videos can be found in the supplementary material provided with this paper, where the changes in texture motion are evident. In the video of flowers fluttering (Fig. 1(a)), in the first 200 frames the flowers are moving very fast, and in the rest they move more slowly. The proposed method correctly detects a change at frame 215. There is an ambiguity of a few frames because the change is detected using $w_0 = 10$ frames around each time instant to approximate the current pdf at frame k, as described in Sec. 3. This ambiguity is negligible, since a difference of about 10 frames is not visible in practice. A video of candles flickering is examined, where the speed of their flickering increases significantly at frame 400. The proposed algorithm detects a change at frame 428. The motion in a video of flamingoes walking becomes much slower at frame 190, and our method finds a change at frame 194. A video of seawater that rain is falling on, first very rapidly, and then more slowly, is examined. The change of the speed of the rain is correctly found at frame 67, as the actual change occurs at frame 70. In a sequence of a toilet flushing, the frame where the water stops running at frame 100 is detected at frame 96, which is very close. Fig. 1(e), (f) show frames 95 and 110, before and after the change, respectively. In the second video, changes are detected at frames 35 when the water starts flowing, and frame 160 when it stops running. Fig. 1(g)-(j) shows frames before and after these changes. In this case false alarms are also found at frames 200 and 220, caused by minor changes in the motion of the trickling water. A video of a barbeque is also examined, where the smoke changes direction and intensity. The change of the smoke's direction is correctly found at frame 82, as seen in Fig. 1(k), (l). The shot change detection of [20] is applied to these videos and, in all cases, is unable to detect the changes, as shown the top half of Table 1.

5.2 Random Motions in Groups

In this section, we examine videos of groups of entities moving together, such as people in crowds, cars and trucks in traffic. Hwere, the change takes place over a few frames rather than at one frame, e.g. a crowd does not enter a scene instantaneously, but over several frames. The ground truth for the correct moment of change is derived by observing the video and is considered to be the central frame of the frame subsequence during which the change takes place.

Highway sequence: varying traffic density: A video of traffic on a highway, whose density changes in the middle of the video, is examined in this experiment. Frame 52 before the change and frame 55 after it are shown in Figs. 2 (a), (b), where the difference in the traffic density is apparent to the observer.

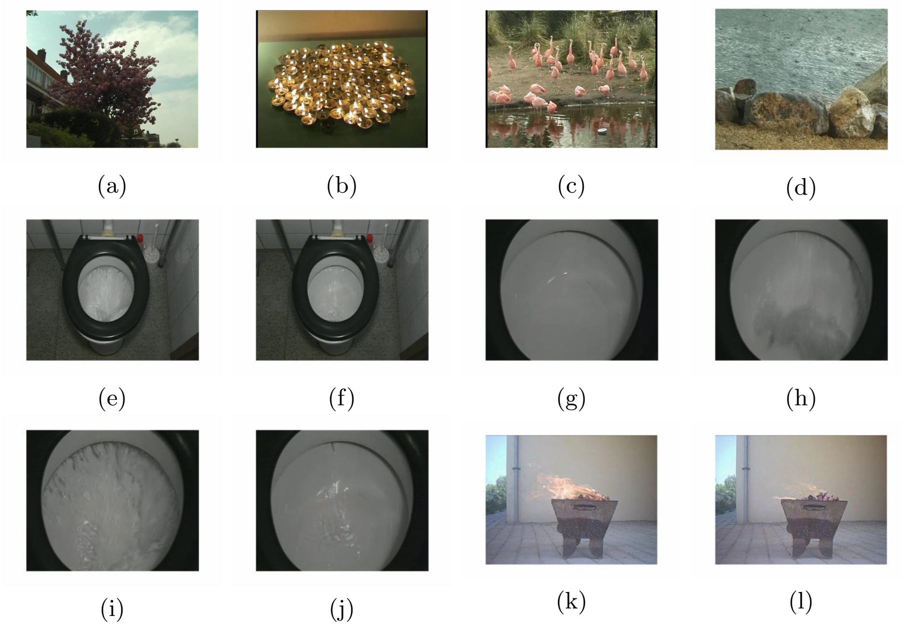

Fig. 1: Temporal textures. (a)-(d): video frames. (e)-(l): before/after a change.

The CUSUM test statistic computed by the proposed method is plotted in Fig. 2 (c), where it is clear that the moment of change in the motion distribution has been captured. This results in the detection of a change at frame 60, i.e. with a small delay of 5 frames.

Traffic sequence: trucks followed by cars: In this video of a highway, initially there are mostly trucks and buses in the scene, followed by traffic consisting mostly of cars. In frame 50 before the change there are indeed mostly cars, and after the change at frame 140, e.g. in frame 180 there are mostly cars (Fig. 2(d), (e)). The CUSUM test statistic computed by the proposed method are plotted in Fig. 2, where it is clear that the moment of change in the motion distribution has been captured almost exactly, at frame 142.

Crowd crossing street sequence: A crowd of people crossing a street is modeled as a random texture. The people approach each other at frame 130 (frame 140 is shown in Fig. 2(g)) and the crowds merge until frame 290, after which they are separated (frame 320 is shown in Fig. 2(h)). The CUSUM of Fig. 2(i) show that changes are indeed correctly detected at frames 140 and 300, which are close to the true change points.

Pedestrians crossing street: A more challenging video showing pedestrians crossing a street with traffic is examined: the changes in the pedestrians' motion are difficult to capture as many people are walking in opposite directions at the same time. The proposed method only detects changes after frame 200

(Fig. 2(l)), which corresponds to a change between many pedestrians to almost none crossing the street. More subtle changes, like the crowds of pedestrians moving in opposite directions merging at frames 70, 150 are not detected, as the motion information extracted from them is not sufficient (Fig. 2(j), (k)). The failure of the algorithm for this sequence is evident from the way the values of the CUSUM test statistic change with time, as shown in Fig. 2(m): these statistics change very little around frame 70, making that change difficult to detect. However, a clear change occurs and is detected after frame 200, after which there are very pedestrians left crossing the street.

Walking sequence: A video of small groups of people appearing, walking in front of a building and exiting is examined. Fig. 2(n)-(p) show the frames where changes were detected, with the groups of people that appear in this video. The CUSUM values of Fig. 2(q) lead to the detection of change at frames 18, when the first group is leaving the scene, at frame 37 when the second group is entering the scene, and frame 59 when they exit the scene. Thus, the proposed method produces correct results for this kind of video as well.

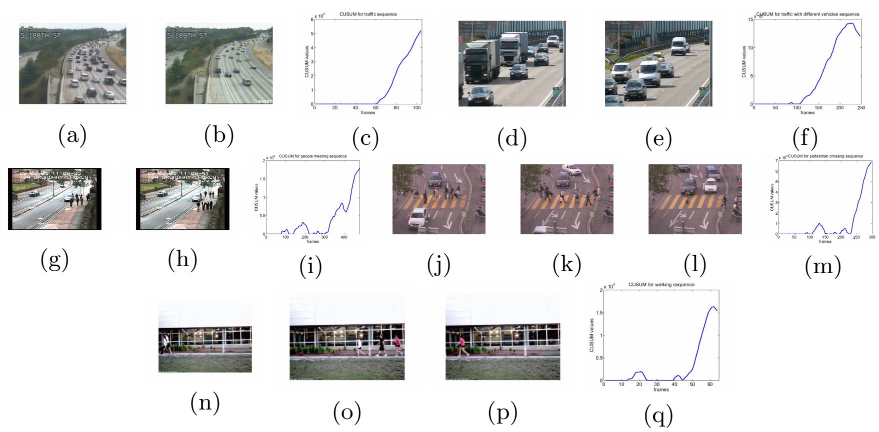

Fig. 2: Random motions in groups. Frames before and after a change in temporal texture, CUSUM Test statistic.

The shot change detection method of [20] is applied to the videos examined here and the results are shown in Table 1 below. The proposed method finds the changes in all cases, except the last video, where only one change is found. However, shot change detection is only able to find a change in the first traffic video, as there is a small change in the scene appearance.

Table 1: Change detection results for Temporal Texture videos.

Videos	Real ch.	Det. ch.	Shot ch. det	Motion before ch.	Motion after ch.
Flowers	200	215	3	Fast motion	Slow motion
Candles	400	428	3	Flicker	Fast flicker
Flamingoes	190	194	3	Fast walk	Normal walk
Water	70	67	3	Fast rain on sea	Normal rain on sea
Toilet 1	100	96	3	No water	Water flushes
Toilet 2	$34, 150$	$35, 160$	3	No water, water flushes	Water flushes, stop
BBQ	90	82	3	BBQ flame, left	flame, right
Traffic 1	53	54	3	Light traffic	Heavy traffic
Traffic 2	140	142	3	Trucks	Cars
Walking	$20, 35, 57$	$18, 37, 59$	3	Exit, enter, exit	Walk,
Crowd	$130, 290$	$140, 300$	476	Meets, part	Walk separately
Pedestrians	$70, 210$	200	297	Peds. meet, stop crossing	No peds.

6 Conclusions

In this work, a novel approach to the segmentation of a video consisting of temporal texture subsequences is presented. The proposed approach provides an approximation to the temporal texture's motion distribution based on properties of the Fourier Transform, leading to a non-parametric model for the motion. Statistical sequential testing, namely the CUSUM test, is then applied to the resulting distributions, in order to detect changes between successive temporal texture subsequences. The need for fitting an appropriate statistical model to the data is avoided, as well as tuning its parameters appropriately. It is necessary to perform empirical testing with training data beforehand, in order to determine a general threshold formula for the categories of videos examined. Experiments show that the proposed method detects the moment of change between temporal texture subsequences with accuracy, based only on their motion characteristics, i.e. without using any appearance information. Comparisons with traditional shot change detection methods show that the CUSUM based approach, with online non-parametric distribution modeling, provides the same or better result, at a much lower computational cost. Shot change detection methods detect changes at the moment of change in sequences where a minor change in appearance has occurred, however they fail in the more challenging videos of crowds of people walking, or of traffic. Future work includes the application of the proposed approach in more complicated videos, where changes are more complex, as well as to videos containing more than one temporal texture in the scene.

Acknowledgements. The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7/2007-2013 under grant agreements: FP7-214306 - JUMAS.

References

1. Rahman, A., Murshed, M.: Temporal texture characterization: A review. Computational Intelligence in Multimedia Processing: Recent Advances (2008) 291–316
2. Polana, R., Nelson, R.: Recognition of motion from temporal texture. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition. (1992) 129–134
3. Szummer, M., Picard, R.: Temporal texture modeling. In: IEEE International Conference on Image Processing (ICIP). (1996) 823826
4. Doretto, G., Chiuso, A., Soatto, S., Wu, Y.: Dynamic textures. International Journal of Computer Vision (IJCV) **51** (2003) 91109
5. Mittal, A., Monnet, A., Paragios, N.: Scene modeling and change detection in dynamic scenes: A subspace approach. Comput. Vis. Image Underst. **113** (2009) 63–79
6. Vidal, R., Ravichandran, A.: Optical flow estimation and segmentation of multiple moving dynamic textures. In: CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2. (2005) 516–521
7. Rahman, A., Murshed, M.: Detection of multiple dynamic textures using feature space mapping. IEEE Transactions on Circuits and Systems for Video Technology **19** (2009) 766 – 771
8. Jacobson, L., Wechsler, H.: Spatio-temporal image processing: Theory and scientific applications. Lecture Notes in Computer Science **751** (1993)
9. Edwards, D., Chang, J.T., Shi, L., Yu, Y.: Motion field estimation for temporal textures. In: Digital Image Computing: Techniques and Applications DICTA. (2003) 389–398
10. Chan, A.B., Vasconcelos, N.: Modeling, clustering, and segmenting video with mixtures of dynamic textures. IEEE Transactions on Pattern Analysis and Machine Intelligence **30** (2008) 909–926
11. Chan, A.B., Vasconcelos, N.: Mixtures of dynamic textures. In: IEEE International Conference on Computer Vision, Proceedings. (2005)
12. Papoulis, A.: Probability, Random Variables, and Stochastic Processes. 2nd edn. McGraw-Hill, New York (1987)
13. Stauffer, C., Grimson, W.: Adaptive background mixture models for real-time tracking. In: Computer Vision and Pattern Recognition, 1999. Proceedings CVPR '99, 1999 IEEE Computer Society Conference on. (1999)
14. Zivkovic, Z., van der Heijden, F.: Efficient adaptive density estimation per image pixel for the task of background subtraction. Pattern Recogn. Lett. **27** (2006) 773–780
15. Page, E.S.: Continuous inspection scheme. Biometrika **41** (1954) 100–115
16. Bershad, N.J., Sklansky, J.: Threshold learning and brownian motion. IEEE Transactions on Information Theory **17** (1971) 350–352
17. Hory, C., Kokaram, A., Christmas, W.: Threshold learning from samples drawn from the null hypothesis for the generalized likelihood ratio cusum test. In: Machine Learning for Signal Processing, 2005 IEEE Workshop on. (2005) 111 – 116
18. Basseville, M., Nikiforov, I.: Detection of Abrupt Changes: Theory and Application. Prentice-Hall, Inc., Englewood Cliffs, N.J. (1993)
19. Muthukrishnan, S., van den Berg, E., Wu, Y.: Sequential change detection on data streams. In: ICDM Workshop on Data Stream Mining and Management, Omaha NE (2007)
20. Chavez, G.C., Cord, M., Foliguet, S.P., Precioso, F., de A. Araujo, A.: Robust scene cut detection by supervised learning. In: EUPISCO. (2006)