Real Time Motion Changes for New Event Detection and Recognition

Konstantinos Avgerinakis, Alexia Briassouli, Ioannis Kompatsiaris

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

Abstract. An original approach for real time detection of changes in motion is presented, for detecting and recognizing events. Current video change detection focuses on shot changes, based on appearance, not motion. Changes in motion are detected in pixels that are found to be active, and this motion is input to sequential change detection, which detects changes in real time. Statistical modeling of the motion data shows that the Laplace provides the most accurate fit. This leads to reliable detection of changes in motion for videos where shot change detection is shown to fail. Once a change is detected, the event is recognized based on motion statistics, size, density of active pixels. Experiments show that the proposed method finds meaningful changes, and reliable recognition.

1 Introduction

Event and activity recognition have become particularly important in the recent years, as they provide valuable information for surveillance, traffic monitoring etc. The video segments processed are usually extracted by shot change detection, or have been segmented before the processing, possibly manually. Shot detection separates the video into subsequences filmed from the same camera/viewpoint and can achieve very high accuracy, but is based on appearance. Activity recognition takes place over video segments that are found by shot change detection in [1]. In practice, this may not always work, as shot detection is based on appearance, although different activities may take place in subsequences with the same appearance. This motivates us to propose a method for separating a video sequence based on motion, which would provide a more meaningful segmentation. Motion has been used for this in [2], where frames with low activity are separated from the others using MPEG-7 motion descriptors, but this is not generally applicable to the case of videos with different activities that need to be separated from each other.

In this work, binary masks of active pixels (Activity Areas) are initially extracted using a kurtosis-based method. The illumination variations over active pixels are processed in the sequel in order to detect changes in them. Statistical modeling of the data shows that the best probability distribution for the sequential likelihood testing is the Laplace. Sequential change detection is then applied to the data to detect changes in it. Since only the currently available video frames are used, the change detection takes place in real time. Once the sequence is divided into subsequences containing different motion, recognition of the action can take place.

This paper is organized as follows. In Sec. 2, the method for extracting the Activity Areas is presented and the CUSUM change detection algorithm is presented in Sec. 3. The statistical modeling required for the CUSUM is included in 3.1. The methods employed for activity/event classification are described in Sec. 4. Experiments with a wide range of indoors and outdoors videos are analyzed in Sec. 5. Finally, conclusions and future work are discussed in Sec. 6.

2 Activity Area

A binary mask of the active pixels in the video, the Activity Area, is helpful in reducing the computational cost of the method and also reducing the possibility of having false alarms, by limiting the data to the truly active pixels. The Activity Area can be extracted at each frame by processing the data available until that moment, i.e. the inter-frame illumination variations until frame k, thus retaining the real time nature of the system. The data at frame k and pixel $\bar{r}$ is a $1 \times k$ vector that can be written as $\mathbf{v}_k(\bar{r}) = [v_1(\bar{r}), ...v_k(\bar{r})]$, where $v_n(\bar{r})$ is the illumination variation at frame $n, 1 \leq n \leq k$, caused either by actual motion or by measurement noise. Each pixel's illumination variation at frame n can be modeled by the following hypotheses:

$$\begin{aligned} H_0 &: v_n(\bar{r}) = z_n(\bar{r}) \\ H_1 &: v_n(\bar{r}) = u_n(\bar{r}) + z_n(\bar{r}), \end{aligned} \tag{1}$$

where $z_n(\bar{r})$ originates from measurement noise and $u_n(\bar{r})$ from actual motion. Additive measurement noise is often modeled as a Gaussian random variable [3], [4], so the active pixels can be discerned from the static ones as they are non-Gaussian. A classical non-Guassianity measure is the kurtosis, which can be employed to separate the active from static pixels, as its value is equal to zero for Gaussian data. For a random variable y, the kurtosis is given by $kurtosis[y] = E[y^4] - 3(E[y^2])^2$. The kurtosis of $\mathbf{v}_k(\bar{r})$ is estimated, to form a "kurtosis mask", which obtains high values at active pixels, and low values at the static ones. The kurtosis has been found to be very sensitive to outliers, and can detect them reliably even for non-Gaussian data [5], [6]. Thus, if the measurement noise deviates from the Gaussian model, the kurtosis will still lead to an accurate estimate of the active pixels. The robustness of the kurtosis for extracting Activity Areas has been analyzed in [7] as well, where it is shown to provide accurate activity areas even for videos with slightly varying backgrounds (e.g. backgrounds with moving trees). The activity areas for some videos used in the experiments in this work are shown in Fig. 1, where it can be seen that the regions of motion are accurately localized. Other foreground extraction methods could also be employed to extract activity areas from the video, such as the

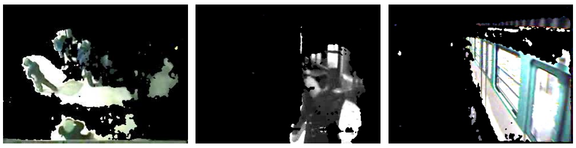

Fig. 1. Activity Areas superposed on frames of videos examined.

Gaussian Mixture models of [8], [9]. The method used should be computationally efficient, like the one proposed here, in order to allow operation in real time.

In practice, there may be errors in an activity area, e.g. a sudden illumination change may cause the entire video frame to be "active". This does not negatively affect the results, since in that case static pixels will also be included in the test, whose flow estimates do not significantly affect the change detection performance. It is also possible that there may be a local occlusion over a few frames that introduces errors in the flow estimates. In most cases, the errors introduced by the occlusion can be overcome because data is collected over a window of frames in which correct (unoccluded) flow values will also be included (see Sec. 3). If, nonetheless, a false alarm is caused by this occlusion, it can be eliminated at a post-processing stage that examines the motion data before and after each change: in the case of false alarms, the motion before and after the false alarm remains the same, so that detected change is ignored.

3 Change Detection

Sequential change detection methods are perfectly suited for designing a real time system for detecting changes, as they are specifically designed for this purpose. Additionally, methods like the CUSUM have been shown to provide the quickest detection of changes in the distribution of a data stream [10], [11]. The data used in this context are the illumination variations of the active pixels in each video frame, which have been extracted using only the currently available video frames. The method used here is the CUSUM (Cumulative Sum) approach developed by Page [12], based on the log-likelihood ratio test statistic at each frame k:

$$T_k = \ln \frac{f_1(\mathbf{V}_k)}{f_0(\mathbf{V}_k)}. \tag{2}$$

Here, $\mathbf{V}_k = [v_1(\bar{r}_1), ..., v_1(\bar{r}_{N_1}), ..., v_k(\bar{r}_1), ..., v_k(\bar{r}_{N_k})]$ represents the illumination of all active pixels over frames 1 to k, assuming that the activity area of each frame n contains N_n pixels. The data distribution before a change is given by $f_0(\mathbf{V}_k)$ and after a change it is $f_1(\mathbf{V}_k)$, so the test statistic of Eq. (2) becomes:

$$T_k = \sum_{i=1}^{k} \sum_{j=1}^{N_i} \ln \frac{f_1(v_i(\bar{r}_j))}{f_0(v_i(\bar{r}_j))}. \tag{3}$$

The log-likelihood ratio uses $\sum_{i=1}^{k} \times \sum_{j=1}^{N_i}$ samples. This is a large number of samples, which provides a good approximation of the data distributions and is expected to lead to reliable detection performance.

In this problem, neither the data distributions before and after a change, nor the time of change are known. In order to find the moment of change using Eq. (2), the distributions f_0 and f_1 have to be approximated. The initial distribution f_0 can be approximated from the first w_0 data samples [13], under the assumption that no changes occur in the first w_0 frames. This is a realistic assumption and does not significantly affect the real time nature of the approach, as errors of 10 frames around a change are almost always difficult to discern visually. The distribution f_1 is approximated at each time instant k using the most recent data available, namely the w_1 most recent frames, in order to avoid a bias towards the baseline pdf f_0. The size of the windows w_0, w_1 is determined by using training data, and it is found that $w_0 = 10,\ w_1 = 1$ led to good distribution approximations and accurate change detection for most videos. These windows are sufficient in size, because they contain all the pixels inside the activity area, which lead to a sufficiently large sample size.

The data is assumed to be independent and identically distributed (i.i.d.) in Eq. 3, an assumption that is common in such problems [14], as joint data distributions can be quite cumbersome to determine in practice. The CUSUM algorithm has been shown to be asymptotically optimal even for data that is not independent [15], so deviations from the i.i.d. assumptions are not expected to introduce noticeable errors. Indeed, in the experiments changes are detected with accuracy under the i.i.d. assumption, under which the test can become computationally efficient, as Eq. 3 obtains the following recursive form:

$$T_k = \max\left(0, T_{k-1} + \ln\frac{f_1(\mathbf{V}_k)}{f_0(\mathbf{V}_k)}\right) = max\left(0, T_{k-1} + \sum_{i=1}^{k}\sum_{j=1}^{N_i} \ln\frac{f_1(v_i(\bar{r}_j)}{f_0(v_i(\bar{r}_j)}\right). \quad (4)$$

The test statistic T_k is compared at each frame with a threshold to find if a change has occurred at that frame. The related literature recommends using training data to find a reliable threshold for good detection performance [11]. We have found that at each time instant k, the threshold can be estimated from $\eta_k = mean([T_{k-1}] + c \times std[T_{k-1}]$, where $mean[T_{k-1}]$ is the mean of the test statistic's values until frame $k-1$ and $std[T_{k-1}]$ is the standard deviation of those values. Very reliable detection results are found for $c = 5$ for the videos used in these experiments.

3.1 Statistical data distribution modeling

The test of Eq. (3) requires knowledge of the family of data probability distributions before and after a change. In the literature, the data has been assumed to follow a Gaussian distribution [3] due to lack of knowledge about its nature. We propose finding a more accurate model for the pdf, in order to achieve optimal detection results. The data under consideration are the illumination variations of each active pixel over time. These variations are expected to contain outliers, as a pixel is likely to be inactive over several frames, and suddenly become active. Data that contains outliers is better modeled by a heavy-tailed distribution, such as the Laplace, the generalized Gaussian or the Cauchy, rather than the Gaussian. We compare the statistical fit achieved by the Laplace and Gaussian distributions, as their parameters can be estimated quickly, without affecting the real time character of the proposed approach. The Laplace pdf is given by:

$$f(x) = \frac{1}{2b} \exp\left(-\frac{|x-\mu|}{b} \right), \tag{5}$$

where μ is the data mean and $b = \sigma/\sqrt{2}$ is its scale, for variance σ^2, which can be directly estimated from the data.

The histogram of the data (illumination variations) is estimated to approximate the empirical distribution. The data mean and variance are also estimated and used to estimate the parameters for the Gaussian and Laplace pdfs. The resulting pdfs are compared both visually and via their mean squared distance from the empirical distribution for the videos used in the experiments. As Fig. 2 shows for several videos, the empirical data distribution is best approximated by the Laplace model. This is expected, since the Gaussian pdf does not account for the heavy tails in the empirical distribution, introduced by the data outliers. The average mean squared error for the approximation of the data by Gaussian and Laplace pdfs is 0.04 for the Laplace distribution, while it is 0.09 for the Gaussian model, verifying that the Laplace is better suited for our data.

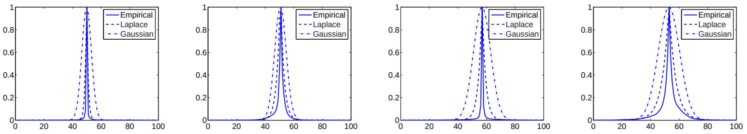

Fig. 2. Statistical modeling using Gaussian, Laplace distributions for traffic videos.

The CUSUM test based on the Laplace distribution then becomes:

$$T_k = \sum_{i=1}^{k} \sum_{j=1}^{N_i} \left(\ln \frac{b_0}{b_1} - \frac{v_i(\bar{r}_j) - \mu_1}{b_1} + \frac{v_i(\bar{r}_j) - \mu_0}{b_0} \right), \tag{6}$$

so the CUSUM test now is:

$$T_k = \max\left(0, T_{k-1} + \sum_{i=1}^{k} \sum_{j=1}^{N_i} \left(\ln \frac{b_0}{b_1} - \frac{v_i(\bar{r}_j) - \mu_1}{b_1} + \frac{v_i(\bar{r}_j) - \mu_0}{b_0} \right)\right) \tag{7}$$

and can be applied to each current data sample after the estimation of the distribution parameters as described in Sec. 3.

4 Recognition

For surveillance videos in various setups, the event of interest focuses on the arrival or departure of people or other entities from the scene. When someone enters a scene, the activity area becomes larger, and when they exit, the activity area size decreases. The experiments show that this leads to correct annotation of such events in a variety of indoors and outdoors scenarios. Additional information can be extracted by examining the velocity before and after a detected change: if it decreases, the activity taking place is slowing down, and may even come to a stop if the velocity after a change becomes zero. Similarly, an increase of speed can easily be detected after a change.

For traffic videos, the events to be recognized are transitions between heavy, medium and light traffic. When there is heavy traffic, the activity area consists of many small connected components, originating from the small vehicle motions. Here, connected component refers to active pixels that are continuous in space, forming coherent groups of pixels. During light traffic, the cars move fast, so the activity areas comprise of fewer connected components. Medium traffic leads to more connected components than light traffic, but fewer than heavy traffic. Training videos of traffic are examined, and it is determined that heavy traffic occurs when there are more than 60 connected components in the activity area, there are $30 - 60$ for medium traffic, and less than 30 for light traffic (Fig. 3). This indeed leads to recognition of the varying traffic conditions, and can be achieved in real time.

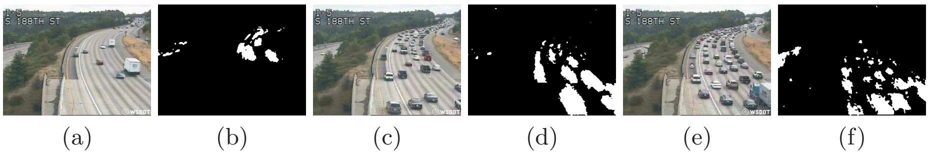

Fig. 3. Light, medium and heavy traffic. The connected components of the active regions increase as the traffic gets heavier.

5 Experiments

Experiments take place with various videos to examine the accuracy of the change detection results, for surveillance and traffic applications. The method is also compared to shot change detection.

Surveillance

A variety of surveillance videos, indoors and outdoors, from banks, entrances, train stations and others, are examined for detection of changes. In all cases, the change points are detected correctly. Figs. 4(a),(b) show the frames before and after a new robber enters to rob an ATM (video duration 1 min 39 sec, at 10 fps). Figs. 4(c),(d) show a security guard before and after he jumps over a gate (video duration 9 sec, at 10 fps). In Figs. 4(e), (f) a train station is shown before and after the train enters, and Figs. 4(g), (h) show the train before and after it slows down (video duration 10 sec, at 10 fps). The examined videos can be seen in the supplementary material, with the moments of change highlighted in red, showing that the changes are correctly detected.

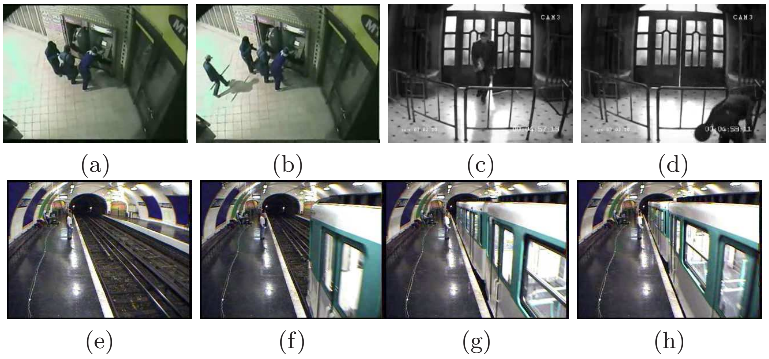
(a) (b) (c) (d)
(e) (f) (g) (h)

Fig. 4. Frames before/after a change. ATM robbery: new robber enters. Guard: guard enters, jumps over gate. Train station: train appears, slows down, stops.

In these videos, recognition consists of finding whether a moving entity is entering or exiting the scene, and if its speed changes. An entrance is detected when the size of the activity area increases in the frames after the detected change, while an exit occurs if the activity area decreases in size. This makes intuitive sense, since more pixels become active as someone enters a scene, and vice versa, and leads to the correct annotation of these events, as can be seen in the supplementary material. Additional information about the activity taking place can be extracted by examining the motion magnitude before and after a change. For the video with the train entering the station, after the fist change, it is found to be slowing down, and after the second change, it stops completely. This method leads to correct annotations that can be seen in the corresponding result videos in the supplementary material. For high-level annotations, additional information about the scene needs to be known, for example context information that the video is of a bank and the location of the ATM can help identify a robbery. Such information can be provided a priori, or extracted from the scene with additional visual processing. In practice, it is likely that contextual information will be available, as a system is designed for a particular application.

Traffic

Traffic videos (of duration 10 sec at 10 fps) are also examined, to detect changes between heavy, medium and light traffic. As can be seen in Fig. 5 the test statistics provide a clear indication of the moment of change. Videos of the highway traffic with these changes highlighted in red are provided in the supplementary material. The proposed method detects the changes correctly: as seen in Fig. 5, the frames before and after the change point clearly contain a different amount of traffic. It finds two changes in the last video, although the last two subsequences in it both contain heavy traffic. This error is introduced because they are filmed in very different weather conditions: the second video is filmed on a rainy day, which changes the motion estimates significantly. However, the recognition stage that follows corrects this false alarm by correctly characterizing both segments as having heavy traffic. The recognition of the type of traffic in each video subsequence takes place based on the number of connected components in the corresponding activity area, as described in Sec. 4, and leads to correct results in all cases.

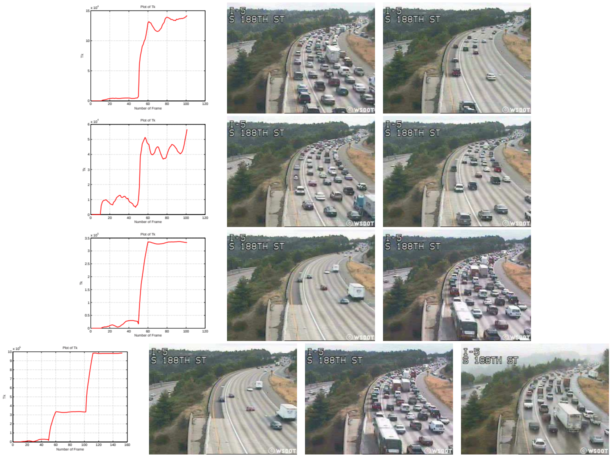

Fig. 5. First column: CUSUM test statistic. Columns 2-4: frames before/after changes.

Comparison with shot change detection

The usefulness of the proposed approach can be better determined when comparing it to traditional shot change detection methods, such as that of [1]. Shot change detection can find changes between shots introduced by variations in appearance, rather than in motion. We apply this method to the videos on which sequential change detection is performed. The ground truth for the changes in motion is found by visually observing the videos. Table 1 shows that the shot change detection is unable to detect most changes in the motion, even when these have caused a slight change in the video appearance. For example, in traffic videos the change from light to heavy traffic may be accompanied by the appearance of more cars in the scene. Nonetheless, shot change detection cannot discern this change, whereas the proposed approach finds it. Similarly, when robbers enter or exit the scene, the proposed method finds these changes, but shot change detection does not. In the last two traffic videos, there are some significant appearance changes that coincide with the motion changes (see supplementary material). These changes are detected by the shot change detection as well, as expected.

Finally, the proposed approach has a lower computational burden than traditional shot change detection. It runs completely in real time, whereas the shot change detection requires several minutes to run on the same videos, in C++. The results for both methods and the corresponding ground truth presented in Table 1 demonstrate that the sequential change detection approach correctly finds frames at which changes occur in the video. This can be used to signal alarms in a security setup, or divide the video into subsequences which can be used at a latter stage as input to an event recognition system.

Table 1. Comparison with shot change detection

Video	True Changes	Our method	Shot ch. det.
ATM robbery	38, 55, 100, 450, 520, 651	42, 58, 102, 458, 530, 654,	-
ATM robbery	685, 729, 790, 814, 891, 908	690, 733, 794 , 818, 896, 913	-
Police Station	20, 35, 100, 140, 167, 210	21, 37, 110, 145, 170, 216	-
Train station	8, 100, 232	10, 104, 237	-
Heavy-Light	50	51	-
Heavy-Medium	50	51	-
Light-Heavy	51	52	51
Light-Heavy-Medium	50, 100	52, 104	103

Acknowledgements: This research has received funding from the European Community's 7th Framework Programme FP7/2007-2013 under grant agreement FP7-214306 - JUMAS.

6 Conclusions

In this work, a novel, real time approach for separating videos into meaningful subsequences with different events is proposed. The active regions of the video are localized using higher order statistics, and a binary mask, the activity area, is produced. Only the motion in pixels inside the activity area is processed, in order to minimize computational cost and probability of false alarms. Sequential change detection, specifically the CUSUM method, is applied to the motion vectors of the video, to detect changes in them in real time. The Laplace model is used to describe the motion vectors, as it accurately describes the outliers in them. Once the video is separated into subsequences containing different activities, recognition is applied to the subsequences to characterize the events taking place in them. The recognition uses information from the activity areas, as well as the motion taking place in them. Comparisons take place with shot change detection, where it is shown that they are unable to detect changes in motion, and therefore different events, which the proposed method can find. Additionally, shot change detection requires significant computational time, whereas the system presented here operates in full time. Experiments with surveillance and traffic videos demonstrate that it provides reliable detection of changes and recognition of the events taking place, making it a reliable tool for numerous applications. Future work includes working with more complex sequences, containing more than one activities which undergo changes.

References

1. Chavez, G.C., Cord, M., Philip-Foliguet, S., Precioso, F., de A. Araujo, A.: Robust scene cut detection by supervised learning. In: EUPISCO. (2006)
2. Ajay, D., Radhakrishan, R., Peker, K.: Video summarization using descriptors of motion activity: a motion activity based approach to key-frame extraction from video shots. J. Electronic Imaging **10** (2001) 909–916
3. Aach, T., Kaup, A., Mester, R.: Statistical model-based change detection in moving video. Signal Processing **31** (1993) 165–180
4. Wren, C., Azarbayejani, A., Darrell, T., Pentland, A.: Pfinder: real-time tracking of the human body. IEEE Transactions on Pattern Analysis and Machine Intelligence **19** (1997) 780–785
5. Hassouni, M., Cherifi, H., Aboutajdine, D.: Hos-based image sequence noise removal. IEEE Transactions on Image Processing **15** (2006) 572–581
6. Giannakis, G., Tsatsanis, M.K.: Time-domain tests for Gaussianity and time-reversibility. IEEE Transactions on Signal Processing **42** (1994) 3460 –3472
7. Briassouli, A., Kompatsiaris, I.: Robust temporal activity templates using higher order statistics. IEEE Transactions on Image Processing (**to appear**)
8. 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)
9. 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
10. Dragalin, V.P.: Optimality of a generalized cusum procedure in quickest detection problem. In: Statistics and Control of Random Processes: Proceedings of the Steklov Institute of Mathematics, Providence, Rhode Island (1994) 107–120
11. Moustakides, G.V.: Optimal stopping times for detecting changes in distributions. Ann. Statist. **14** (1986) 13791387
12. Page, E.S.: Continuous inspection scheme. Biometrika **41** (1954) 100–115
13. 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)
14. Lelescu, D., Schonfeld, D.: Statistical sequential analysis for real-time video scene change detection on compressed multimedia bitstream. IEEE Transactions on Image Processing **5** (2003) 106–117
15. Bansal, R.K., Papantoni-Kazakos, P.: An algorithm for detecting a change in a stochastic process. IEEE Transactions on Information Theory **32** (1986) 227–235