Real Time Illumination Invariant Motion Change Detection

ABSTRACT

An original approach for real time detection of changes in motion is presented, which can lead to the detection and recognition of events. Current video change detection focuses on shot changes, which depend on appearance, not motion. Changes in motion are detected in pixels that are found to be active via the kurtosis. Statistical modeling of the motion data shows that the Laplace distribution provides the most accurate fit. The Laplace model of the motion is used in a sequential change detection test, which detects the changes in real time. False alarm detection determines whether a detected change is indeed induced by motion or by varying scene illumination. This leads to precise detection of changes in motion for many videos, where shot change detection is shown to fail. Experiments show that the proposed method finds meaningful changes in real time, even under conditions of varying scene illumination.

Categories and Subject Descriptors

[**Content Track**]: Video Segmentation, Video surveillance, Event detection

Keywords

change detection, motion analysis, illumination invariance

1. INTRODUCTION

Event and activity recognition have become particularly important in the recent years, as they can provide valuable information for surveillance, medical monitoring, smart homes, traffic monitoring etc. The video segments being processed are usually extracted by shot change detection, or have been segmented before the processing, possibly manually. Today, shot detection can achieve very high accuracy for any kind of shot change, be it abrupt or gradual, but it is based on similarity of appearance. Object trajectories obtained over the entire video are processed in [9] as well as in [11] for event detection. Activity recognition takes place over video segments that are found by shot change detection in [17]. In practice, this approach may not always work, as different activities may take place in subsequences with the same appearance, i.e. in subsequences that belong to the same shot. 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 extracted at each frame, 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 based on this model, in order to detect changes in it. A false alarm test is implemented after each change is detected, to determine if they are true or false alarms introduced by scene illumination variations. Since only the currently available video frames are used, the change detection takes place in real time.

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. A method for acquiring robustness to scene illumination variations is presented in Sec. 4, and the proposed procedure for change verification is presented in Sec. 4.1. Experiments with a wide range of indoors and outdoors videos are analyzed in Sec. 6, including videos with global scene illumination variations. Finally, conclusions and future work are discussed in Sec. 7.

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 being processed to the truly active pixels. The Activity Area can be extracted at each frame by using 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

ACM-MM 2010 Firenze, Italy

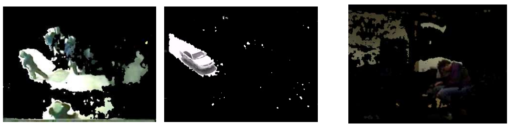


Figure 1: Activity Areas superposed on frames of videos examined.

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:

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

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 [1], [16], 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 $kurtosis[y] = E[y^4] - 3(E[y^2])^2$, which obtains the value zero when y is Gaussian. The kurtosis is also defined as

$$kurtosis[y] = \frac{E[y^4]}{(E[y^2])^2} - 3,$$

which is equal to zero for Gaussian data. Then, 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. It should be noted that the kurtosis has been found to be very sensitive to outliers, and can detect them reliably even for non-Gaussian data [8], [7]. Thus, if the measurement noise deviates from the Gaussian model, the kurtosis is still expected to lead to an accurate estimate of the active pixels. The robustness of the kurtosis for extracting Activity Areas has been analyzed in [4] 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 Gaussian Mixture models of [15], [18]. However, the method used should be computationally efficient, like the one proposed here, in order to allow operation in real time.

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 [6], [12]. 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 [14], based on the log-likelihood ratio test statistic at each frame k:

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

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:

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

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. The distribution f_1 is approximated at each time instant k using the data available until that moment. Since it should describe the data after a change, it is found using 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.

The data is assumed to be independent and identically distributed (i.i.d.) in Eq. 3, an assumption that is common in such problems [10], 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 [3], 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. As shown in [14], the test obtains a computationally efficient recursive form for i.i.d. data, as Eq. 3 becomes:

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

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 [12]. We have found that at each time instant k, the threshold can be estimated from:

$$\eta_k = mean([T_{k-1}] + c \times std[T_{k-1}], \qquad (5)$$

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 accurate detection results are found for $c = 5$ for the videos used in these experiments.

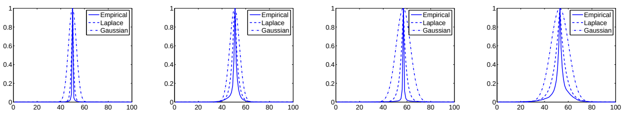


Figure 2: Statistical modeling using Gaussian, Laplace distributions for surveillance videos.

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 [1] 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. In this work we compare the statistical fit achieved by the Laplace and Gaussian distributions, as their parameters can be estimated very quickly, without affecting the real time character of the proposed approach. The Laplace pdf is given by:

$$f(x) = \frac{1}{2b} \exp\left(-\frac{|x-\mu|}{b}\right), \tag{6}$$

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.

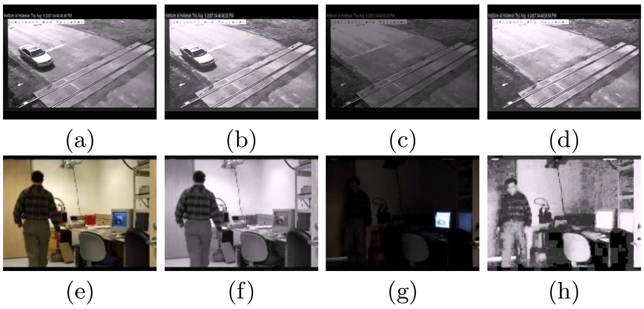
The CUSUM test based on the Laplace distribution then becomes:

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

so the CUSUM test now is:

$$\begin{aligned} T_k \;=\; & \max\Bigg(0, T_{k-1} + \\ & \sum_{i=1}^{k} \sum_{j=1}^{N_i} \left(\ln \frac{b_0}{b_1} - \frac{v_i(\bar{r}_j) - \mu_1}{b_1} + \frac{v_i(\bar{r}_j) - \mu_0}{b_0} \right)\Bigg) \end{aligned} \tag{8}$$

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


(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 3: Frames before and after equalization. (a), (b), (e), (f) Frames before the illumination variation. (c), (d), (g), (h) Frames after the illumination variation.

4. ROBUSTNESS TO ILLUMINATION VARIATION

In real videos, it is possible to have significant illumination variations that can be mistaken for changes. In indoors videos these can be caused by turning a light on and off, in outdoors they may originate from a shadow falling over the camera, the sun being covered by clouds and many other factors. The proposed method is based on the motion information in the video, so it is expected to provide detection results that are relatively robust to such scene illumination changes. Nevertheless, these changes translate into motion vectors, which may lead to false alarms, especially in the case of a significant global illumination change.

In order to increase the system's robustness to illumination variations, we propose to apply histogram equalization to each video frame before testing for a change. This is expected to reduce sensitivity to global illumination variations, as it will reduce the contrast in each frame and distribute the grayscale values more equally among the pixels. After histogram equalization is applied to each pair of frames, their optical flow is computed and sequential change detection proceeds as described in the previous sections. Figs. 3(a), (b) show how equalization affects a frame before global scene illumination changes and Figs. 3(c), (d) show an original and equalized frame after the scene illumination has changed. It is clear that the effect of equalization is to more evenly distribute the illumination values in the video frames. In the case of the darker image, the even redistribution of its values reduces the amount of dark pixels, so the resulting equalized frame is similar to the one before the actual illumination change. A very large scene illumination change caused by a person turning a light off in the office is shown in Figs. 3(e)-(h). The effect of the equalization here is quite striking, as the person is barely visible in the original image, but can be clearly seen after it is equalized. From these examples, it is expected that this preprocessing step will significantly help in avoiding scene illumination change induced false alarms.

4.1 Change verification

In practice, false alarms may appear even after histogram equalization, so a method for detecting false alarms is proposed. After each change is detected, the average illumination before a change is compared with the average illumination after it. For a video frame of $N = N_1 \times N_2$ pixels, where $l(\bar{r}(n), k)$ is the pixel intensity for the n_{th} pixel $\bar{r}(n)$, the average scene illumination over w frames before and after frame k is given by $l_0 = \frac{1}{wN}\sum_{i=k-w+1}^{k}\sum_{n=1}^{N} l(\bar{r}(n), i)$ and $l_1 = \frac{1}{wN}\sum_{i=k+1}^{k+w}\sum_{n=1}^{N} l(\bar{r}(n), i)$ respectively. The ratio of the average scene illuminations is given by:

$$\rho_l = \frac{\max(l_1, l_0)}{\min(l_1, l_0)}, \tag{9}$$

and is compared to the threshold of $\eta = 1.1$ in order to detect global illumination changes. For noiseless data, a threshold value of 1 would indicate global scene illumination changes. However, in practice there may be small illumination variations caused by measurement noise or other sources of noise. For this reason, a slightly higher threshold of 1.1 is chosen, which indeed provides reliable false alarm detection for a wide range of both indoors and outdoors videos. Then, if ρ_l is above η, the detected change is attributed to a change in the global scene illumination and is removed from the list of detected changes. In that case, the amount of global illumination variation that has taken place is known and is equal to ρ_l. Once a change caused by global illumination variation is found, that change point is eliminated, since it does not correspond to a change in motion. The change detection procedure needs to be reset at that time instant, which will now be considered as the first frame, since the data before and after it has a different scene illumination and will always (falsely) appear to have undergone a change. This renders the proposed method robust to scene illumination variations, making it useful for practical applications like surveillance, where the lighting in the scene under examination changes.

5. ALGORITHM OVERVIEW

In this section we present an overview of the proposed algorithm's steps, corresponding to the previous sections and the block diagram of Fig. 4.

1. Acquire video frame pairs and apply histogram equalization to each frame.

2. Compute optical flow for current frames.

3. Extract Activity Area from current frames.

4. Estimate Laplacian parameters for initial and current data.

5. Apply sequential change detection testing via the CUSUM test and the current data.

6. Verify if a change is true or a false alarm caused by changing scene illumination.

7. Separate video into subsequences of coherent activity if a change is true, and apply sequential change detection to the rest of the video.

6. EXPERIMENTS

Experiments take place with various videos to examine the accuracy of the change detection results. Sec. 6.1 presents results for videos where the illumination remains constant, and Sec. 6.2 presents results for videos where the illumination changes, either artificially or naturally.

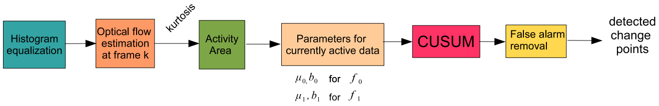
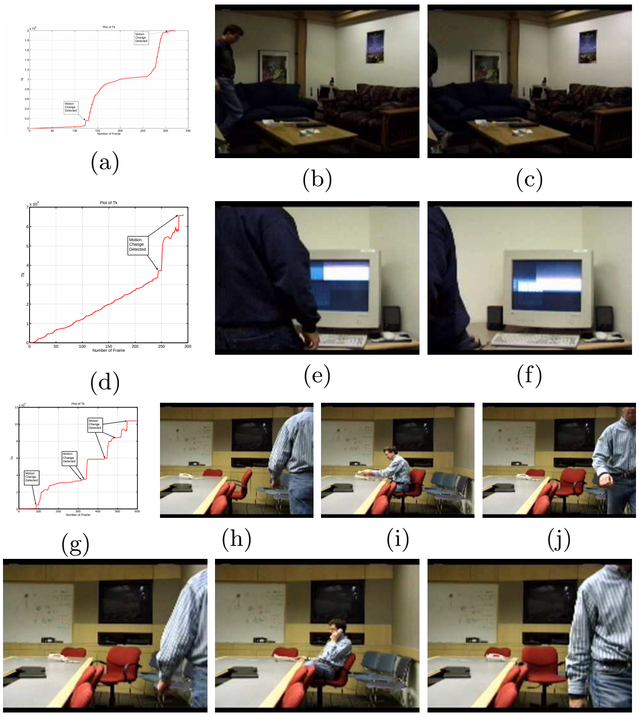


Figure 4: Block diagram of proposed system.



(a) (b) (c) (d) (e) (f) (g) (h) (i) (j) (k) (l) (m)

Figure 5: Test statistic and frames before and after a change, in the presence of illumination changes. (a)-(c): Person enters room with computer. (d)-(f): Person enters a room to read, leaves. (g)-(m): Person enters conference room, leaves.

6.1 Videos with constant illumination

A video of a person entering a room, sitting to read and then getting up and leaving is examined. The proposed method detects a change when the person enters, and when he gets up to leave, while there are no false alarms (Figs. 5(a)-(c)). Another video, of a person entering a room where a computer is on, is examined. Here, the computer screen is flickering, but this does not introduce any false alarms. A change is detected when the person enters the room, and when he exits. The test statistic is shown in Fig. 5(d), where is can be seen that it indeed undergoes a significant increase at the moments of change. This is also verified by the frames of change, shown in Figs. 5(e), (f). Finally, a video of a conference room where a person enters and exits is examined. As Figs. 5(g)-(m) show, the changes are correctly captured.

6.2 Videos with varying illumination

In this section,the performance of the proposed method for videos with varying illumination is examined. In the first video, the illumination is artificially darkened in its last part. The proposed method detects a change when the car starts moving and when it exits the scene, but does not detect a false alarm at the frame where the global illumination is modified. Fig. 6(a) shows the corresponding test statistic, which indeed increases suddenly at the frames where the changes occur, and the frames of change are shown in Fig. 6(b), (c).

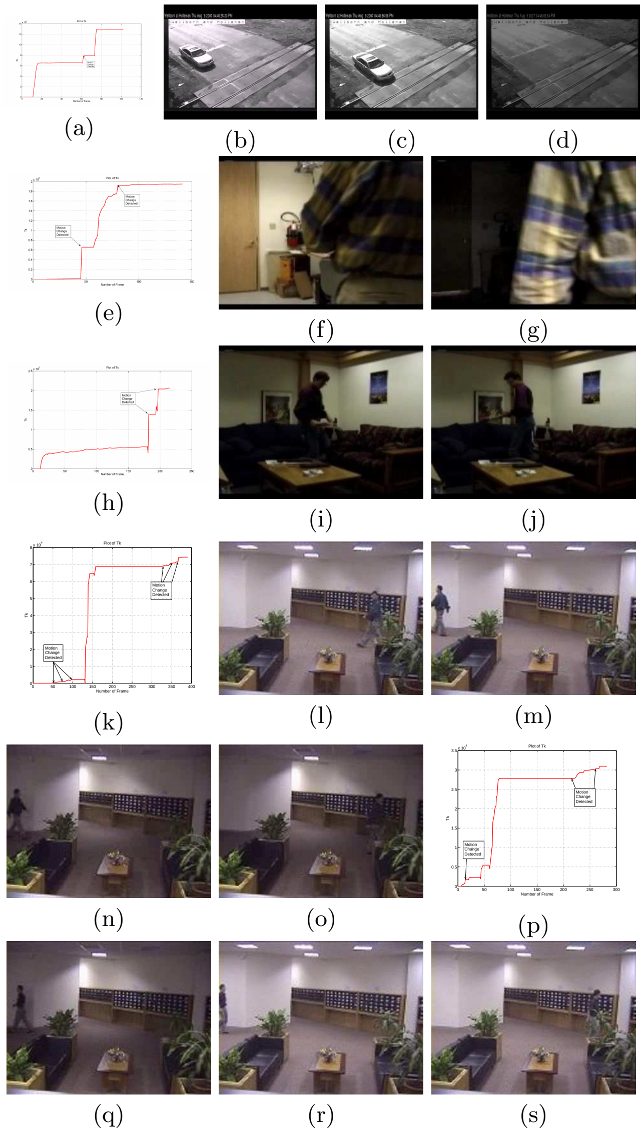
In the next video, a person enters an office, turns the light off, and exits. After histogram equalization, a change is detected when the person enters and when he exits. The test statistic that led to these changes is shown in Fig. 6(e) and the corresponding frames are shown in Fig. 6(f), (g). Similar results are reported for the other videos examined. Results for a person entering a room, sitting to read, and getting up to leave are shown in Figs. 6(h)-(j). In the first frames, there are gradual illumination changes, however they are not mistaken for changes in motion. Two videos of a lobby where the light is turned on and off gradually, throughout all the frames, are examined. In both videos, the motion changes are correctly detected, while the false alarms introduced by the illumination variations are all successfully eliminated. Figs. 6(k)-(s) show the test statistic and the frames of change for the two videos. Changes are found when the person enters or exits the area. It can be seen in these figures that the scene illumination is different in different parts of each video.

6.3 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 [5]. 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, after applying histogram equalization to their frames. This is done to ensure fairness of comparison with our method, since the scene illumination changes are diminished after histogram equalization. We find that the shot change detection is unable to detect most changes in motion, but tends to find changes caused by the illumination variations, even after equalization of the video frames. It should also be emphasized that the proposed approach 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 are presented in Tables 1 for videos with constant illumination. In this case, the shot change detection detects a true change only for the video where the person enters the room with the computer. This can be easily explained by the fact that the person's body covers the entire frame as he enters, so it significantly changes the scene appearance. In the other cases, only our method is successful in finding the changes.

In videos with varying scene illumination, shot change detection is unable to find the change detected by our approach in the video of the car at the railroad crossing. It also misses the scene illumination change, as it has been eliminated by the video equalization. In the video of the person entering the room to turn off the light, both methods find the changes. This is expected as the appearance changes significantly when the person enters and exits the video, as he covers the entire video frame. Finally, the gradually varying



(a) (b) (c) (d)

(e) (f) (g)

(h) (i) (j)

(k) (l) (m)

(n) (o) (p)

(q) (r) (s)

Figure 6: Test statistic and frames before and after a change, in the presence of illumination changes. (a)-(d): Car at railroad crossing. (e)-(g): Person enters room, turns light off, exits room. (h)-(j): Person enters room while light is gradually turned on, sits, leaves.

illumination is not mistaken for a change in motion in the video where a person enters a room and sits to read. Our method finds the changes in his motion, but the shot change detection misses them. For the videos of the lobby with the varying illumination, the changes caused by a person entering and exiting are successfully detected by our method. The shot change detection only detects the moments where the scene illumination changes. Naturally, this erroneous detection can also be eliminated by the change verification step that our method uses.

Table 1: Comparison with shot change detection in the absence of illumination changes

Video	True Changes	Our method	Shot det.
Read	122, 295	124, 300	-
Office computer	241, 281	244, 283	243

Table 2: Comparison with shot change detection in the presence of illumination changes

Video	True Changes	Our method	Shot det.
Railroad	61	62	-
Turn off light	46, 79	46, 80	46, 72
Read, var. ill.	180, 195	182, 196	-
Lobby 1	48, 98, 323, 371	51, 96, 320, 367	154
Lobby 2	18, 217, 265	14, 215, 261	60

7. CONCLUSIONS

In this work, a novel, real time approach for separating videos into meaningful subsequences, that is robust to changing scene illumination, is proposed. The active regions of the video are localized using higher order statistics, and a corresponding 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, and specifically the CUSUM method, is applied to the motion vectors of the video, in order 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. After a change is detected, it is examined to determine whether or not it is a false alarm using an appropriately designed test. The resulting changes are detected in real time and the overall system is robust to global illumination changes. Comparisons take place with shot change detection techniques, where it is shown that they are unable to detect changes in motion, and therefore different events, which the proposed method can find. Shot change detection techniques are also computationally costly, unlike the proposed system, which operates in full time. Experiments with surveillance and monitoring videos demonstrate that it provides accurate detection of changes, even in the presence of illumination variations, making it a reliable tool for numerous applications. Future work includes developing methods for recognition of the activities taking place in the resulting subsequences.

8. REFERENCES

[1] T. Aach, A. Kaup, and R. Mester. Statistical model-based change detection in moving video. *Signal Processing*, 31(2):165–180, 1993.

[2] D. Ajay, R. Radhakrishan, and K. Peker. Video summarization using descriptors of motion activity: a motion activity based approach to key-frame extraction from video shots. *J. Electronic Imaging*, 10(4):909–916, 2001.

[3] R. K. Bansal and P. Papantoni-Kazakos. An algorithm for detecting a change in a stochastic process. *IEEE Transactions on Information Theory*, 32(2):227–235, 1986.

[4] A. Briassouli and I. Kompatsiaris. Robust temporal activity templates using higher order statistics. *IEEE Transactions on Image Processing*, to appear.

[5] G. C. Chavez, M. Cord, S. Philip-Foliguet, F. Precioso, and A. de A. Araujo. Robust scene cut detection by supervised learning. In *EUPISCO*, 2006.

[6] V. P. Dragalin. Optimality of a generalized cusum procedure in quickest detection problem. In *Statistics and Control of Random Processes: Proceedings of the Steklov Institute of Mathematics*, pages 107–120, Providence, Rhode Island, 1994.

[7] G. Giannakis and M. K. Tsatsanis. Time-domain tests for Gaussianity and time-reversibility. *IEEE Transactions on Signal Processing*, 42(12):3460 –3472, Dec. 1994.

[8] M. Hassouni, H. Cherifi, and D. Aboutajdine. Hos-based image sequence noise removal. *IEEE Transactions on Image Processing*, 15(3):572–581, 2006.

[9] A. Hervieu, P. Bouthemy, and J.-P. L. Cadre. A statistical video content recognition method using invariant features on object trajectories. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(11):1533–1543, 2000.

[10] D. Lelescu and D. Schonfeld. Statistical sequential analysis for real-time video scene change detection on compressed multimedia bitstream. *IEEE Transactions on Image Processing*, 5(1):106–117, 2003.

[11] G. Medioni, I. Cohen, F. Bremond, S. Hongeng, and R. Nevatia. Event detection and analysis from video streams. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(8):873–889, Aug. 2006.

[12] G. V. Moustakides. Optimal stopping times for detecting changes in distributions. *Ann. Statist.*, 14:1379Ű1387, 1986.

[13] S. Muthukrishnan, E. van den Berg, and Y. Wu. Sequential change detection on data streams. In *ICDM Workshop on Data Stream Mining and Management*, Omaha NE, Oct. 2007.

[14] E. S. Page. Continuous inspection scheme. *Biometrika*, 41:100–115, 1954.

[15] C. Stauffer and W. Grimson. Adaptive background mixture models for real-time tracking. In *Computer Vision and Pattern Recognition, 1999. Proceedings CVPR '99, 1999 IEEE Computer Society Conference on*, June 1999.

[16] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder: real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):780–785, 1997.

[17] L. Xie, D. Xu, S. Ebadollahi, K.Scheinberg, S. Chang, and J. Smith. Detecting generic visual events with temporal cues. In *Signals, Systems and Computers, 2006. ACSSC '06. Fortieth Asilomar Conference on*, pages 54–58, 2006.

[18] Z. Zivkovic and F. van der Heijden. Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern Recogn. Lett.*, 27(7):773–780, 2006.